

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería de Tecnologías y Servicios de la
Telecomunicación

TRABAJO FIN DE GRADO

**HERRAMIENTA PARA IDENTIFICAR
CONVALIDACIONES DE ASIGNATURAS EN
PROGRAMAS DE INTERCAMBIO**

Miguel Vila Rodríguez
Tutor: Francisco Saiz López
JUNIO 2020

HERRAMIENTA PARA IDENTIFICAR CONVALIDACIONES DE ASIGNATURAS EN PROGRAMAS DE INTERCAMBIO

AUTOR: Miguel Vila Rodríguez
TUTOR: Francisco Saiz López

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2020

Resumen (castellano)

A la hora de preparar mi experiencia personal de Erasmus me fue complicado encontrar un destino con convalidaciones para las asignaturas que iba a cursar, y pensé que se podía hacer algo para facilitar y reducir la carga de esta tarea. Este trabajo ha consistido en la creación y composición de una herramienta para facilitar la identificación de esas convalidaciones.

Se trata de una aplicación web que se conecta a una base de datos en la cual se guardan los registros de convalidaciones para Erasmus de años anteriores. Existen tres tipos de registros (relacionados entre sí): universidades, asignaturas y convalidaciones.

La herramienta permite dos roles: administrador y usuario estándar. Ambos pueden visualizar los datos existentes en la base de datos, buscar registros por columnas, así como ordenarlos ascendente o descendientemente. El primero además se encargará de gestionar la herramienta, insertando y modificando registros si fuera necesario.

El trabajo ha implicado la creación por medio de SQL de una base de datos, a la cual se han cargado los archivos Excel de convalidaciones de Erasmus de la web de la EPS mediante funciones (macros) VBA. La aplicación se conecta a la base de datos para visualizar, insertar o modificar los registros. En cuanto a la aplicación en sí, se ha dividido en dos. Primero, la parte del servidor o Backend, que se conecta a la base de datos gracias a los lenguajes Java y JSP, y tecnologías adicionales como AJAX. Conectada al servidor, la parte del cliente o Frontend, realizada gracias a HTML5, CSS3 y JavaScript, destacando jQuery como la biblioteca más usada de esta última.

Abstract (English)

When I was preparing for my personal Erasmus experience, I found it difficult to find a destination with validations for the subjects that I was going to take, and I thought something could be done to make this task easier and simpler. This project consisted in the creation and composition of a tool to facilitate the identification of those validations.

It is made up of a web application that is connected to a database in which the records of validations for Erasmus from previous years are stored. There are three types of records (related to each other): universities, subjects and validations.

The tool allows two roles: administrator and standard user. Both can view the existing data in the database, search for records by columns, as well as sort them in ascending or descending order. However, the first one will also be in charge of managing the tool, inserting and modifying records if necessary.

The project has involved the creation of a database using SQL, to which the Erasmus validations' Excel files of the EPS website have been uploaded using VBA functions (macros). The application connects to the database to visualize, insert, or modify the records. As for the application itself, it has been divided in two parts. First, the server part or Backend, which connects to the database thanks to the Java and JSP languages, and additional technologies such as AJAX. Connected to the server, the client part or Frontend, developed thanks to HTML5, CSS3 and JavaScript, highlighting jQuery as the most used library of the latter.

Palabras clave (castellano)

Erasmus, Herramienta, Base de datos (BD), Universidades, Asignaturas, Convalidaciones, Aplicación web, Servidor, Java, JSP, Cliente, HTML5, CSS3, JavaScript, jQuery, AJAX, DataTables.

Keywords (inglés)

Erasmus, Tool, Database, Universities, Subjects, Validations, Web Application, Server, Java, JSP, Client, HTML5, CSS3, JavaScript, jQuery, AJAX, DataTables.

Agradecimientos

Me gustaría agradecer a todas las personas involucradas en este proyecto, directa o indirectamente.

En primer lugar, me gustaría darle las gracias a mi tutor Francisco Saiz por escucharme y apoyar mi idea desde el principio, cuando apenas estaba definida.

Muchas gracias al estudiante que participó en la entrevista que figura al final de esta memoria. Sus opiniones fueron útiles para obtener una idea de cómo mejorar la herramienta ahora, y en el futuro.

Gracias a mi familia por ser mi constante apoyo en la vida. No habría llegado aquí si no fuera por la energía, la mentalidad y la educación (entre otras) que me han transmitido desde pequeño.

Gracias a los profesores que me han transmitido gran parte de los conocimientos que me han servido para elaborar este proyecto.

He de agradecer su ayuda indirecta a todos los usuarios de Internet que distribuyen sus conocimientos de forma gratuita. También a los usuarios de Stack Overflow, sitio web donde he recibido ayuda inmediata cuando he preguntado alguna duda sobre programación.

También quería darle las gracias a la persona que rellena los Excel de la web de la EPS, ya que han sido una buena fuente para comprobar el buen funcionamiento de la herramienta.

Por último, y no menos importante, gracias también a la UAM en general, por estos años tan buenos que he pasado en sus instalaciones, llenándome de conocimiento y de aprendizaje, no solo a nivel curricular, sino también a nivel personal.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Organización de la memoria.....	2
2	Estado del arte.....	3
2.1	Introducción.....	3
2.2	Sistemas de gestión de bases de datos (SGBD).....	3
2.2.1	Oracle.....	3
2.2.2	MySQL.....	3
2.2.3	Microsoft SQL Server.....	4
2.3	Aplicaciones web.....	5
2.3.1	Sección del servidor (Backend Web).....	5
2.3.2	Sección del cliente (Frontend web).....	8
2.3.3	Datos de las aplicaciones web.....	11
2.4	Microsoft Visual Basic for Applications.....	12
3	Diseño.....	13
3.1	Tecnologías empleadas.....	13
3.1.1	Base de datos.....	13
3.1.2	Aplicación web.....	17
3.2	Esquema completo de la herramienta.....	22
4	Desarrollo.....	23
4.1	Roles de usuario.....	23
4.1.1	Administradores.....	23
4.1.2	Usuarios estándar (alumnos).....	23
4.2	Funcionalidades de la aplicación web.....	24
4.2.1	Página inicial.....	24
4.2.2	Inserción de registros.....	25
4.2.3	Visualización de registros.....	28
4.2.4	Modificación de registros.....	33
4.2.5	Registros inactivos.....	34
4.3	Control de errores.....	35
4.4	Seguridad.....	38
5	Integración, pruebas y resultados.....	39
5.1	Pruebas técnicas.....	39
5.2	Entrevista a futuro estudiante de Erasmus.....	39
6	Conclusiones y trabajo futuro.....	41
6.1	Conclusiones.....	41
6.2	Trabajo futuro.....	41
	Referencias.....	43
	Glosario.....	45
	Anexos.....	- 1 -
A	Manual de instalación.....	- 1 -
B	Ficheros de parámetros.....	- 3 -

INDICE DE FIGURAS

FIGURA 1: RANKING DE LOS SGBD DE SOFTWARE LIBRE MÁS CONOCIDOS [5].....	4
FIGURA 2: COMPARATIVA DE ALGUNOS SGBD [3]	4
FIGURA 3: POSIBLE ESTRUCTURA DE APLICACIÓN WEB [8].....	5
FIGURA 4: COMPARATIVA DE PHP Y JAVA [11]	6
FIGURA 5: FLUJO DE SOLICITUD/RESPUESTA AL LLAMAR A UN JSP [13].....	6
FIGURA 6: ESQUEMA CLIENTE-SERVIDOR PARA SERVLETS (PARTE 1) [14].....	7
FIGURA 7: ESQUEMA CLIENTE-SERVIDOR PARA SERVLETS (PARTE 2) [14].....	7
FIGURA 8: LOGO DE ANGULAR.....	9
FIGURA 9: LOGO DE REACT	10
FIGURA 10: LOGO DE VUE.JS	10
FIGURA 11: LOGO DE JQUERY	10
FIGURA 12: LOGO AJAX.....	11
FIGURA 13: LOGO DE MICROSOFT VBA	12
FIGURA 14: CAMPOS DE LA TABLA <i>UNIVERSITIES</i>	15
FIGURA 15: CAMPOS DE LA TABLA <i>SUBJECTS</i>	15
FIGURA 16: CAMPOS DE LA TABLA <i>CONVALIDATIONS</i>	15
FIGURA 17: LOCALIZACIÓN DE LOS FICHEROS EXCEL EN LA WEB DE LA EPS	16
FIGURA 18: EJEMPLO DE REGISTRO DE LA TABLA <i>UNIVERSITIES</i>	16
FIGURA 19: EJEMPLO DE ALGUNOS REGISTROS DE LA TABLA <i>SUBJECTS</i>	16
FIGURA 20: EJEMPLO DE ALGUNOS REGISTROS DE LA TABLA <i>CONVALIDATIONS</i>	16
FIGURA 21: ORGANIZACIÓN DE LOS FICHEROS JSP EN LA APLICACIÓN	18
FIGURA 22: FICHEROS JAVA DE LA APLICACIÓN.....	19
FIGURA 23: ARCHIVOS HTML DE LA APLICACIÓN	19
FIGURA 24: ORGANIZACIÓN DE LOS FICHEROS CSS EN LA APLICACIÓN WEB.....	20
FIGURA 25: ORGANIZACIÓN DE ALGUNOS FICHEROS Y BIBLIOTECAS JAVASCRIPT	21

FIGURA 26: EJEMPLO DE USO DE JAVASCRIPT + AJAX PARA MOSTRAR LA TABLA DE CONVALIDACIONES	21
FIGURA 27: ESQUEMA COMPLETO DE LA HERRAMIENTA DISEÑADA. REALIZADO GRACIAS AL SITIO WEB CANVAS.....	22
FIGURA 28: PÁGINA INICIAL DE LA APLICACIÓN WEB.....	24
FIGURA 29: MENÚ DESPLEGABLE PARA USUARIOS ESTÁNDAR	24
FIGURA 30: MENÚ DESPLEGABLE PARA ADMINISTRADORES	24
FIGURA 31: FORMULARIO DE LA APLICACIÓN WEB PARA AÑADIR UNIVERSIDADES	25
FIGURA 32: FORMULARIO DE LA APLICACIÓN WEB PARA AÑADIR ASIGNATURAS	25
FIGURA 33: FORMULARIO DE LA APLICACIÓN WEB PARA AÑADIR CONVALIDACIONES.....	26
FIGURA 34: EJEMPLO DE INSERCIÓN CORRECTA DE DATOS (I).....	27
FIGURA 35: EJEMPLO DE INSERCIÓN CORRECTA DE DATOS (II).....	27
FIGURA 36: VISUALIZACIÓN DE UNIVERSIDADES EN LA APLICACIÓN WEB PARA USUARIOS ESTÁNDAR	28
FIGURA 37: VISUALIZACIÓN DE ASIGNATURAS EN LA APLICACIÓN WEB PARA USUARIOS ESTÁNDAR	29
FIGURA 38: VISUALIZACIÓN DE CONVALIDACIONES EN LA APLICACIÓN WEB PARA USUARIOS ESTÁNDAR	29
FIGURA 39: VISUALIZACIÓN DE CONVALIDACIONES EN LA APLICACIÓN WEB PARA ADMINISTRADORES	29
FIGURA 40: POSIBILIDAD DE ELEGIR DISTINTO NÚMERO DE FILAS A MOSTRAR.....	30
FIGURA 41: EJEMPLO DE BÚSQUEDA DE "CIRCUITOS ANALÓGICOS" EN LA TABLA DE CONVALIDACIONES	30
FIGURA 42: EJEMPLO DE BÚSQUEDA DE "124" EN LA TABLA DE CONVALIDACIONES	30
FIGURA 43: EJEMPLO DE BÚSQUEDA DE "ANTENAS PISA" EN LA TABLA DE CONVALIDACIONES	31
FIGURA 44: COPIA DE 3 REGISTROS AL PORTAPAPELES	31
FIGURA 45: COMPARATIVA DEL ESPACIO QUE OCUPA CADA TIPO DE ARCHIVO.....	31
FIGURA 46: PDF GENERADO CON LA BÚSQUEDA DE "ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL"	32
FIGURA 47: CSV GENERADO CON LA BÚSQUEDA DE "ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL"	32

FIGURA 48: EXCEL GENERADO CON LA BÚSQUEDA DE "ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL"	32
FIGURA 49: PÁGINA DE EDICIÓN DE UN REGISTRO DE UNIVERSIDADES	33
FIGURA 50: PÁGINA DE EDICIÓN DE UN REGISTRO DE ASIGNATURAS	33
FIGURA 51: PÁGINA DE EDICIÓN DE UN REGISTRO DE CONVALIDACIONES	34
FIGURA 52: MENSAJE DE MODIFICACIÓN CORRECTA.....	34
FIGURA 53: CORRECCIÓN DE ERRORES DE LA BD	36
FIGURA 54: COMPROBACIÓN DE CAMPO VACÍO (I).....	37
FIGURA 55: COMPROBACIÓN DE CAMPO VACÍO (II).....	37
FIGURA 56: COMPROBACIÓN DE ERRORES DE RANGO EN FORMULARIO	37
FIGURA 57: CÓDIGO JAVASCRIPT DE COMPROBACIÓN DE CAMPOS VACÍOS	37

INDICE DE TABLAS

TABLA 1: EJEMPLO DE CONVALIDACIÓN MIXTA O CON ASIGNATURAS MÚLTIPLES	26
---	----

1 Introducción

1.1 Motivación

Todo el proyecto surge de mi experiencia personal como alumno de Erasmus en el curso 2019/20. Me encontré un poco perdido a la hora de buscar asignaturas de universidades extranjeras que pudiera convalidar por asignaturas de la EPS, sobre todo al tratarse de asignaturas de cuarto curso de mi grado. Pese a que la web dispone de información y ficheros Excel al respecto que sirven de gran utilidad, al principio es muy costoso empezar a buscar. Hay que elegir un destino el cual sea de nuestro agrado y que a la vez tenga un número aceptable de asignaturas que hayan sido convalidadas en otros años, ya que el objetivo del Erasmus es estudiar un año fuera de una manera productiva, sin perder año/s de carrera. Se trata de un proceso que nos puede llevar mucho tiempo. Muchos alumnos lo ven como algo muy complicado, y aunque se dispone de un largo plazo para realizar estas convalidaciones, es frecuente desistir y prolongar la decisión mucho tiempo.

Gracias a la tecnología y la facilitación que esta es capaz de otorgar, pensé en la posibilidad de amenizar un poco más esta tarea a las generaciones venideras. El desarrollo de esta herramienta gira en torno a esta misión. Estudiar fuera y conocer un poco el mundo es un bien muypreciado para algunos, y es por ello por lo que la mejora de las condiciones para hacerlo es algo beneficioso.

1.2 Objetivos

El objetivo principal de este TFG consiste en desarrollar una herramienta que ayude a los alumnos a encontrar convalidaciones de asignaturas de una forma más sencilla. A su vez, permite dos roles: administrador y alumno. Buscamos una herramienta que:

- Permita a ambos tipos de usuarios visualizar los registros de convalidaciones, universidades y asignaturas.
- Facilite a ambos tipos de usuarios la búsqueda manual mediante palabras clave, para filtrar registros y encontrar resultados más rápidamente.
- Permita la edición a los administradores. Esto incluye la posibilidad de insertar, borrar y/o modificar registros.
- Sea capaz de cargar y almacenar de manera óptima los registros de los archivos Excel de la web de la EPS.
- Controle errores a la hora de la inserción de datos por parte de los administradores.

Además de los propios objetivos de la herramienta, enfoco este trabajo como un reto personal en el cual hay otras metas, añadidas a las del propio trabajo.

- Aprender nuevos lenguajes, tecnologías y métodos de programación y la combinación entre ellos.
- Poner en práctica mis conocimientos de Java y SQL

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1: Introducción.** Se tratan conceptos generales. Incluye la motivación del trabajo y los objetivos a cumplir.
- **Capítulo 2: Estado del arte.** En este capítulo se explicarán conceptos y tecnologías que se han examinado y tenido en cuenta para llevar a cabo el desarrollo de la herramienta, incluyendo las existentes en la propia herramienta.
- **Capítulo 3: Diseño.** A partir de las tecnologías disponibles, analizaré en profundidad el diseño de la herramienta y los motivos existentes para realizarlo de esa forma.
- **Capítulo 4: Desarrollo.** Se explicarán en detalle todas las funcionalidades de la herramienta, así como los aspectos técnicos a tener en cuenta para llegar a conocer bien cómo funciona.
- **Capítulo 5: Integración, pruebas y resultados.** Se expondrán algunas pruebas que se han realizado para conseguir un funcionamiento adecuado de la herramienta, los resultados de estas y como han influido en la realización del proyecto. También habrá una pequeña entrevista a un futuro estudiante de Erasmus que probó la herramienta, para conocer realmente su utilidad y tener una opinión subjetiva.
- **Capítulo 6: Conclusiones y trabajo futuro.** En esta sección figurarán las conclusiones surgidas de todo el proceso de diseño, creación, desarrollo y prueba y error de la herramienta.
- **Anexos.** Se incluyen algunos anexos que sirven de complemento a este trabajo. En ellos se indica qué archivos deben ejecutarse para crear la base de datos y cargar en ella los datos. También se indica cómo se puede ejecutar la aplicación web para poder ser probada. Además, se explica qué parámetros se deben cambiar para personalizar la herramienta a gusto del instalador.

2 Estado del arte

2.1 Introducción

Para consultar convalidaciones de Erasmus en años anteriores actualmente existe la opción de recorrer de forma manual los Excel existentes en la web de la EPS. El objetivo de este proyecto es facilitar esta tarea de consulta. Para ello desarrollaremos una herramienta de ayuda, la cual será una aplicación web que se conectará a una base de datos. En este capítulo, primero se describirán los sistemas de gestión de bases de datos más conocidos. Posteriormente, se expondrán las aplicaciones web, y las formas más comunes y útiles de desarrollarlas, así como el proceso y el modo de obtención de datos. Por último, se describirá brevemente el lenguaje de macros VBA, el cual será necesario para cargar los Excel de la web de la EPS a la base de datos.

2.2 Sistemas de gestión de bases de datos (SGBD)

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en un conjunto de datos interrelacionados y un conjunto de programas para almacenar, acceder, modificar y extraer esos datos [1]. Estos programas se desarrollan con lenguaje SQL (Structured Query Language), el cual fue creado por Ted Codd a principios de la década de 1970 con un modelo relacional basado en IBM. Se volvió cada vez más popular y acabó reemplazando a lenguajes similares de aquella época [5]. Posteriormente se crearon los SGBD para manejar las instrucciones SQL. Centraremos la atención en los SGBD de software libre principales, ya que son los que tengo a mi alcance y que, por tanto, podré utilizar.

2.2.1 Oracle

La compañía fue fundada en 1977 principalmente gracias a un estudio sobre los SGBD llevado a cabo por George Koch. Sorprendieron al mundo con su filosofía de bases de datos relacionales, la cual era algo novedoso en ese momento [4]. Actualmente cuentan con múltiples ediciones, siendo la gran mayoría de ellas de pago excepto la Express Edition (XE). Esta edición es de software libre, pero tiene un almacenamiento y procesamiento limitado de la base de datos.

2.2.2 MySQL

MySQL fue desarrollado por la compañía sueca MySQL AB en 1994. Más tarde, esta compañía fue adquirida por Sun Microsystems en 2008, la cual fue finalmente adquirida por Oracle en 2010 [5]. El software MySQL® proporciona un servidor de base de datos SQL muy rápido y robusto, multi-threaded y multi usuario. Está diseñado tanto para entornos con alta carga de datos y de trabajo como para integrarse en software destinado a ser distribuido [2]. La mayor ventaja de MySQL es precisamente su software libre sin ningún tipo de límite de almacenamiento o de CPU, además de su alta velocidad a la hora de acceder a los datos.

2.2.3 Microsoft SQL Server

Otro SGBD importante es Microsoft SQL Server. El código original fue desarrollado en los años 80 por Sybase Inc., pero posteriormente se unieron al proyecto Microsoft y Ashton-Tate Corp., entonces el principal proveedor de bases de datos para PC, produciendo la primera versión de lo que se convirtió en Microsoft SQL Server, en 1989. En 1994, Microsoft se hizo cargo de todo el desarrollo y la comercialización de SQL Server para sus propios sistemas operativos, y en 1996, Sybase renombró a su versión como Adaptive Server Enterprise, dejando el nombre de SQL Server a Microsoft [6]. La principal ventaja de este gestor es su baja complejidad para el usuario, pero al igual que Oracle, está limitado a un almacenamiento y procesamiento determinados.

Rank	Rank			DBMS	Database Model	Score		
	Dec 2018	Nov 2018	Dec 2017			Dec 2018	Nov 2018	Dec 2017
1.	1.	1.	1.	Oracle +	Relational DBMS	1283.22	-17.89	-58.32
2.	2.	2.	2.	MySQL +	Relational DBMS	1161.25	+1.36	-156.82
3.	3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1040.34	-11.21	-132.14
4.	4.	4.	4.	PostgreSQL +	Relational DBMS	460.64	+20.39	+75.21
5.	5.	5.	5.	MongoDB +	Document store	378.62	+9.14	+47.85

Figura 1: Ranking de los SGBD de software libre más conocidos [5]





free database software	Platform	Limited to	Ease of use	Cloud Version
Altibase 	Linux	No limitation	Very Easy	Yes
MySQL 	Windows, Linux, Mac.	No limitation	Easy	Yes
Oracle 	Windows, Linux	1 GB RAM 11 GB database. 1CPU.	Medium	Yes
SQL Server 	Windows, Linux.	1 GB RAM & 10 GB database. 1 CPU.	Very Easy	No

Figura 2: Comparativa de algunos SGBD [3]

2.3 Aplicaciones web

Las aplicaciones web consisten en un software estilo cliente-servidor. La información se visualiza a través de un navegador web, que será el lado del cliente. Cuando en este navegador el usuario pincha sobre algún enlace hipertexto, estableceremos una petición de un archivo residente en el servidor (un ordenador continuamente conectado a la red). El navegador recibe el archivo e interpreta su información, obteniendo así la información que el usuario ha solicitado.

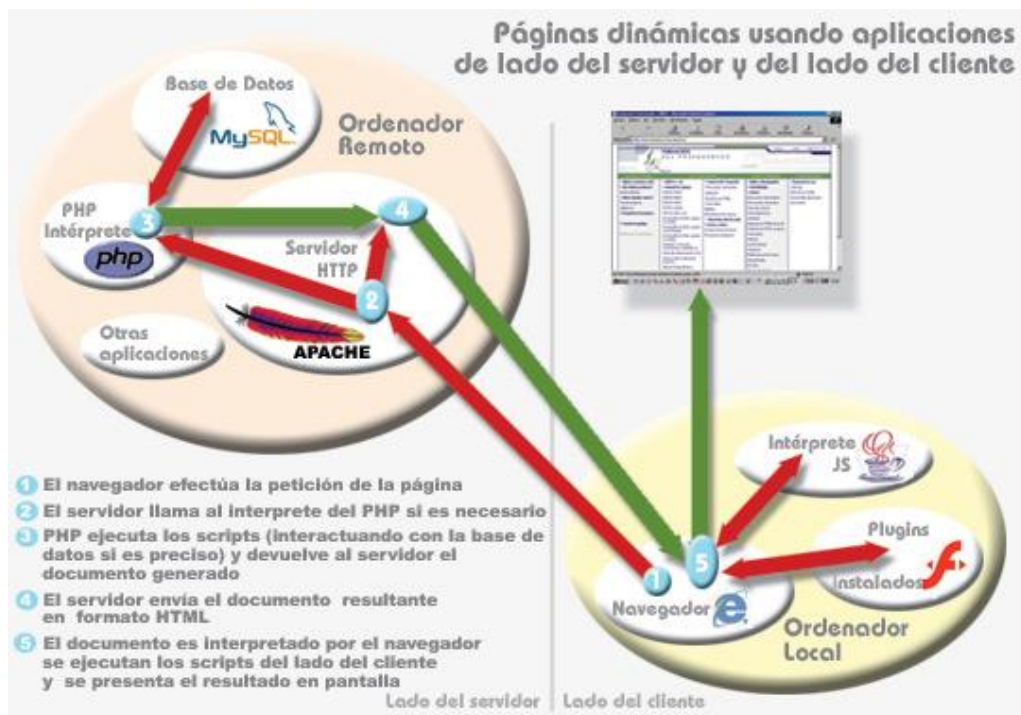


Figura 3: Posible estructura de aplicación web [8]

De esta manera, existen tecnologías, lenguajes y técnicas para el lado del servidor (también denominada la parte de Backend Web) gracias a las cuales los datos son reconocidos, interpretados y ejecutados por el propio servidor y son enviados al cliente en un formato inteligible para este. Asimismo, las herramientas del lado del cliente (equivalente a la parte de Frontend Web) pueden ser reconocidas e interpretadas por la mayoría de los navegadores directamente, sin necesidad de un procesamiento.

2.3.1 Sección del servidor (Backend Web)

Es importante conocer las tecnologías disponibles para llevar a cabo el desarrollo de la parte de Backend Web: será el sector de la aplicación encargado de manejar los datos y proveer con información a la parte de Frontend.

La opción más extendida internacionalmente es la que une un lenguaje de programación con una base de datos, de manera que mediante las funciones de ese lenguaje combinadas con queries SQL se puedan extraer, modificar, eliminar... los registros de la base de datos.

Para esta combinación, posiblemente el lenguaje más famoso y favorito de muchos programadores web es **PHP** (acrónimo recursivo de Hypertext Processor). Es un lenguaje multiplataforma y de código abierto que puede ser fácilmente adaptable con HTML5.

Otra opción bastante conocida y prácticamente igual de válida que PHP es **Java**. Es uno de los lenguajes utilizados desde hace más tiempo para programación Backend. Es cierto que para aplicaciones que se desea comercializar o poner en línea, Java puede tener un coste mayor que PHP, pero en cuanto a la portabilidad, rendimiento o la cantidad de soporte y ayuda disponible son lenguajes muy similares. Hablando de seguridad, Java incluso proporciona un nivel mayor [11].

	PHP	Java
Costos	✓ Más barato	✗ Más Costoso
Rendimiento	✓ Buen Rendimiento	✓ Buen Rendimiento
Seguridad	✗ Menos seguro	✓ Más seguro
Sintáxis	✓ Amigable	✗ Compleja
Portabilidad	✓ Portable	✓ Portable
Tiempo de Desarrollo	✓ Poco tiempo	✗ Más tiempo
Soporte y Ayuda	✓ Buen Soporte y Ayuda	✓ Buen Soporte y Ayuda

Figura 4: Comparativa de PHP y Java [11]

Dentro de las tecnologías que utilizan Java destacan las **Java Server Pages (JSP)**, las cuales combinan HTML con expresiones, directivas, scriptlets o etiquetas de Java, entre otras. Cuando se llama a una página JSP, esta se compila en un Java Servlet (módulos Java que ayudan a extender las capacidades de los servidores web), el cual es manejado por el motor de servlets. Este motor carga la clase y la ejecuta creando un HTML dinámico que se manda al navegador.

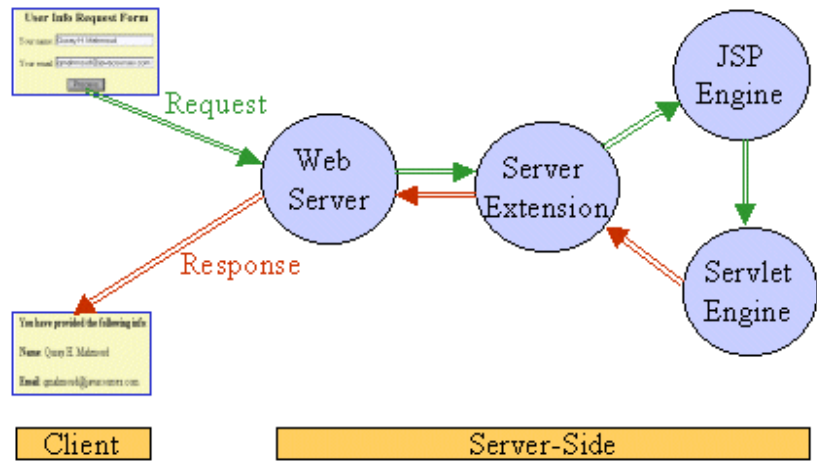


Figura 5: Flujo de solicitud/respuesta al llamar a un JSP [13]

Destacan también los propios **Servlets**, con Java como uno de sus motores principales. Son programas desarrollados en Java que se ejecutan en el servidor web. Cuando este recibe una petición del cliente (en nuestro caso un usuario sobre el navegador), el Servlet lleva a cabo una acción. Las principales acciones que puede llevar a cabo son:

- Generar una respuesta HTML
- Llamar a otros servlets
- Utilizar cookies y sesiones
- Crear un enlace entre un cliente y una base de datos

El esquema simple que siguen los Servlets es el siguiente:



Figura 6: Esquema cliente-servidor para Servlets (parte 1) [14]



Figura 7: Esquema cliente-servidor para Servlets (parte 2) [14]

Como se puede observar, el Servlet genera “de la nada” un documento HTML, y a partir de ello una respuesta al cliente. Para llevar a cabo esto, primero se debe crear una clase que herede de `HttpServlet` para conectarse al cliente y recibir la petición, para después establecer el formato de respuesta HTML gracias a las funciones de clase, y finalmente poder generarla. El funcionamiento de los Servlets es parecido a JSP. Por lo general se utiliza JSP cuando la lógica de la aplicación no es muy extensa, pero si el código a tratar es complejo y grande conviene utilizar Servlets, y combinarlos con JSP, de manera que haya cooperación y comunicación entre ambos. [14]

Por otro lado, **Python** se ha convertido también en una elección popular para programación Backend, debido a la simpleza de su sintaxis y a la amplia cantidad de librerías de las que dispone. También cabe mencionar otro lenguaje bastante utilizado, conocido como **Ruby**. Fue desarrollado en Japón, y ha sido bastante popular debido a su sencillez e intuición. Sin embargo, mucha gente lo critica y se comenta que cada vez está desapareciendo más.

Otra posibilidad destacable para el desarrollo del Backend es mediante entornos de desarrollo web que facilitan esta tarea. El más importante es **ASP.Net**, perteneciente a Microsoft y sobre el que se suele programar en VB script (Visual Basic script) o en C#.

Alternativamente al uso de los navegadores, existe la posibilidad de utilizar entornos de ejecución para JavaScript. Entre ellos destacan **Node.js** y **Deno**, ambos contruidos basándose en el motor de JavaScript V8 de Chrome. Es importante comentar que estas herramientas deben gran parte de su utilidad a la existencia de sus frameworks o paquetes, entre los que destacan Express, Koa y Hapi, entre otras.

Por último, y en el primer puesto en cuanto a la automatización del Backend, existen ciertas tecnologías que proporcionan de manera automática el Backend, que constituyen el modelo denominado **Backend as a service (BaaS)**, también conocido como Mobile BaaS (MBaaS). Este modelo permite vincular las aplicaciones a la nube para almacenar información o bases de datos, o a servicios analíticos para la aplicación, entre otras posibilidades. Destaca su uso para aplicaciones multiplataforma. Algunos de los proveedores de Baas principales son: Backendless, Back4app, Firebase, Azure, Oracle Cloud... La mayoría de ellos ofrecen un servicio básico gratuito, el cual deja de serlo si el desarrollador desea añadir más funcionalidades. Baas ofrece múltiples ventajas en cuanto a rendimiento y portabilidad de una aplicación web. Sin embargo, cuanto más automatizamos la programación del lado del servidor, menos personalizada será nuestra aplicación, lo cual es un rasgo que hay que tener en cuenta [16].

2.3.2 Sección del cliente (Frontend web)

La parte de Frontend web es la que tendrá contacto con el usuario, por lo que es necesario utilizar de manera correcta los lenguajes disponibles. La gran mayoría de los desarrolladores web utilizan la combinación de HTML, CSS y JavaScript, aunque también se utilizan otros lenguajes como VB Script, Active X, applets de Java... y tecnologías que necesitan un plug-in en el navegador como por ejemplo Adobe Acrobat Reader o Autodesk MapGuide, entre otras. [9]

En primer lugar, HTML (HyperText Markup Language), fue desarrollado en 1992 para crear la estructura básica de páginas web, organizar su contenido y compartir información.

En la actualidad se utiliza **HTML5**, que es la última actualización de HTML y un término para agrupar al propio HTML, a CSS3 y JavaScript, ya que las aplicaciones web ya no solo describen el lenguaje sino también diseño y funcionalidad. [17][18]

CSS (Cascading Style Sheet) se utiliza como complemento de HTML para superar las limitaciones de este en cuanto al diseño y reducir su complejidad. Debido a la evolución de HTML y su incapacidad para satisfacer las demandas de los diseñadores, se adoptó CSS como la forma de separar estructura y presentación. Actualmente se utiliza la versión 3, **CSS3**, la cual continúa por este camino, pero ahora teniendo más en cuenta la integración con HTML. CSS trabaja junto con HTML otorgándole estilos visuales al documento: tamaño, color, fondo, bordes... Esto se realiza gracias a etiquetas y referencias a los elementos del documento HTML. [17]

Por último, pero no menos importante, está el famoso lenguaje conocido como **JavaScript**. Es un lenguaje de programación que se utiliza para las funcionalidades y acciones de aplicaciones web o páginas web dinámicas. Es por ello por lo que puede ser considerado como uno de los pilares más fuertes de HTML5. Muchos navegadores incorporan por defecto interfaces de programación de aplicaciones (APIs) para asistir a JavaScript en sus funciones, de manera que se facilita la creación de programas para la aplicación. Es un lenguaje realmente útil, ya que se puede incluir dentro de HTML si se desea, y además tiene un gran alcance debido a la multitud de librerías, APIs y frameworks que lo sustentan [17]. Entre las librerías y frameworks principales se encuentran jQuery, Google Polymer, D3.js, React, Vue.js, Pixi.js, Three.js, Angular, SWFObject...[22]. Hemos de centrarnos solo en la descripción de algunas ya que la lista es considerablemente extensa:

- **Angular:** se trata de un framework bastante conocido, probablemente debido a que está mantenido por Google. Está diseñado para facilitar el diseño de aplicaciones web, reduciendo mucho el trabajo del programador creando una arquitectura limpia y ordenada para su aplicación. Cuando se creó utilizaba exclusivamente JavaScript. Sin embargo, ahora está desarrollado en TypeScript, un lenguaje algo más amplio y general que JavaScript. Algunos usuarios critican su complejidad, y es realmente un framework sin mucho éxito entre los expertos de JavaScript. Pese a ello, muchas empresas confían en este framework debido a la gran comunidad que ha generado Google tras él. [19]



Figura 8: Logo de Angular

- **React:** normalmente se habla de React como una librería, pero a veces los desarrolladores se refieren a él como framework. Es sencillo de entender, declarativo y flexible, de manera que permite construir interfaces de usuario complejas y atractivas. Cuenta con un importante respaldo y mantenimiento por parte de Facebook y algunas otras compañías, por lo que su uso está bastante extendido. [19]

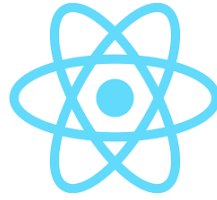


Figura 9: Logo de React

- **Vue.js:** es uno de los frameworks más importantes, al igual que React. Se trata de un framework de código abierto que posee un número considerable de plantillas que podemos utilizar para acelerar el desarrollo de nuestra aplicación. [19]



Figura 10: Logo de Vue.js

- **jQuery:** creada por John Resig, contiene bibliotecas de código JavaScript, que nos permiten interactuar más fácilmente con los documentos HTML, simplificar la manera de manejar el árbol DOM (árbol proveniente de un documento HTML que muestra sus elementos y la interconexión entre ellos), y proporcionar interacción con la tecnología AJAX, que explicaremos más adelante. [20] Es importante mencionar en este apartado a **jQuery UI**. Se trata de una interfaz de usuario (UI) construida sobre jQuery que incluye múltiples iconos, efectos y elementos gráficos para añadir a nuestra aplicación. [21]



Figura 11: Logo de jQuery

Dentro de jQuery es fundamental hablar de algunas de las herramientas relacionadas con esta librería. Una de ellas es **Bootstrap**, una tecnología diseñada originalmente basándose en jQuery, que utiliza librerías CSS, y que permite utilizar elementos ya desarrollados como botones, menús, tipografías... Otra de estas herramientas de jQuery es **DataTables**. Se trata de un plug-in realmente útil que es capaz de enviar solicitudes al servidor mediante AJAX, recibir los datos en formato JSON y mostrar la información en tablas que pueden ser ordenadas por columnas, que permiten realizar búsquedas, y que incluso pueden ser exportadas a archivos externos, como PDFs o CSVs.

Como un derivado de JavaScript también es fundamental hablar de la tecnología **AJAX**. Sus siglas significan Asynchronous JavaScript And XML. Se trata de una combinación de tres elementos: un servidor, el lenguaje JavaScript y un objeto XMLHttpRequest (XHR) perteneciente a un navegador. El objeto XHR actúa como puente para recibir y lanzar peticiones desde y hacia el servidor de varias formas, siendo los métodos GET y POST, de jQuery, los más comunes. El servidor devolverá la respuesta, que puede venir en varios tipos de formato (texto, XML de respuesta, JSON...). Será procesada también por JavaScript y enviada al cliente. Como se puede ver, AJAX no llega a ser una tecnología totalmente del lado de Frontend, pero tampoco se puede considerar Backend ni mucho menos. Podría decirse que se sitúa entre ambas secciones y actúa como vehículo transportador de peticiones, respuestas y, en definitiva, datos. [23]



Figura 12: Logo AJAX

AJAX es una herramienta utilizada a gran escala por muchas compañías punteras como por ejemplo Google o Amazon, ya que gracias a ella las aplicaciones web toman un aspecto muy cómodo y parecido a las aplicaciones de escritorio. Las páginas no necesitan ser recargadas al completo, por ejemplo, para mostrar el carrito de la compra de Amazon o el mapa cuando Google Maps nos enseña la ruta más rápida. Esto es así porque al tratarse de una tecnología asíncrona se pueden realizar varias peticiones de manera simultánea, y así no es necesario esperar a hacer una petición hasta que la anterior haya sido procesada. [23]

En resumen, con la extensa variedad de tecnologías disponibles a nuestro alcance para desarrollar las secciones de Frontend y Backend de una aplicación web, es conveniente meditar y tener en cuenta todas las posibilidades para saber cuáles se adaptan mejor a nuestro caso y a nuestra comodidad.

2.3.3 Datos de las aplicaciones web

Otra sección fundamental de las aplicaciones web son los datos. La gran mayoría de aplicaciones web no funcionarían sin los datos que se procesan y envían. Podría clasificarse de manera muy general los métodos por los cuales se reciben y tratan los datos; habría la posibilidad de cargarse cada vez que se inicia la aplicación web o bien la de almacenarse en una base de datos, a la cual estaría conectada el servidor.

En la opción más sencilla, si cargásemos los datos cada vez que iniciásemos la aplicación y/o recargásemos una página, esta se consideraría estática, y simplemente utilizaría un código fijo con los datos que requiriese nuestra aplicación, con la posibilidad de visualizarlos, pero no la de modificarlos. Esta opción está dirigida a sitios web como portfolios, blogs o páginas web de empresas pequeñas y medianas.

En cambio, si se utiliza una base de datos, el procesado y las opciones para tratar estos es mucho mayor. Al almacenar los datos en registros situados en tablas ordenadas pertenecientes a una base de datos, es mucho más complejo el camino que siguen estos datos. Cuando un usuario solicita unos datos determinados, el navegador se los entrega gracias a que el servidor se los proporciona. Esta entrega se realiza a través de AJAX y/o

JavaScript, normalmente. En el caso inverso, cuando el usuario modifica algún dato en la sección de Frontend, las modificaciones se entregan al servidor. A su vez, el servidor se conecta a la base de datos para obtener o modificar la información de los registros mediante queries o solicitudes. Para poder llevar a cabo estas queries, debe haber una conexión al gestor de base de datos (SGBD) que se está utilizando, lo cual se suele hacer gracias a un controlador o **driver JDBC (Java Database Connectivity)** para servidores Java u **ODBC (Open Database Conectivity)** para otros lenguajes o aplicaciones. Otros tipos de servidores, por ejemplo, los desarrollados en PHP, no requieren expresamente de un driver para lograr esta conexión. Hoy en día incluso están disponibles las bases de datos en la nube para muchos tipos de aplicaciones, pero también dependerá del modo en que vayamos a desarrollar nuestro servidor. [24][25]

Es importante conocer bien nuestra aplicación y los datos que va a manejar, de manera que seamos capaces de diseñar de una manera eficaz el almacenamiento y procesamiento de estos, así como la forma en que se muestran al usuario.

2.4 Microsoft Visual Basic for Applications

Una vez realizado el análisis completo de las aplicaciones web, considero importante explicar otra de las tecnologías que son necesarias para la realización de este trabajo. Hay que tener en cuenta que, como he comentado anteriormente, se han tomado como referencia para la base de datos los ficheros Excel de la página web de la EPS. Es por ello por lo que necesitamos alguna forma de leer esos ficheros y guardar su información, y la forma más lógica de hacerlo es mediante programas o scripts de algún tipo. Estos serán desarrollados en el lenguaje conocido como Microsoft Visual Basic for Applications.

Microsoft VBA es un lenguaje de macros (conjuntos de instrucciones) de Microsoft que sirve para programar y ampliar las funciones de las aplicaciones pertenecientes a esta compañía, y por ello está incluido en la mayor parte de ellas. Destacan Excel, Word o Access entre muchas otras. Yo centraré brevemente el análisis en Excel, ya que es la aplicación que necesito manejar, pero la metodología es similar para la mayoría de las aplicaciones de Microsoft.

VBA es un lenguaje bastante intuitivo y que cuenta con múltiples librerías compuestas por funciones realmente útiles y que reducen mucho el tiempo de desarrollo. Combinando estas funciones con una buena organización de nuestras macros podemos llegar a conseguir programas que automatizan tareas como abrir múltiples ficheros Excel y recorrerlos enteros, almacenando sus registros correctamente en una base de datos. [26]



Figura 13: Logo de Microsoft VBA

3 Diseño

Habiendo listado y explicado las posibilidades que existen para desarrollar una herramienta como la que se presenta en este trabajo, es momento de explicar cuáles de esas tecnologías se han escogido y cómo es la combinación entre ellas. En este punto de la memoria entraremos a fondo a conocer la composición y descripción de cada elemento presente en la herramienta.

3.1 Tecnologías empleadas

A gran escala, la herramienta está compuesta por una aplicación web conectada a una base de datos. Sin embargo, es necesario explicar de manera intensiva el contenido de ambas secciones para comprender bien el esquema final del trabajo.

3.1.1 Base de datos

Es una parte fundamental de la herramienta, ya que es donde se almacenan los registros de asignaturas convalidadas. Fue la primera parte sobre la cual se meditó, ya que sin datos la aplicación servía de poco y era importante diseñar la estructura correctamente para poder manejarla de una manera eficaz y cómoda.

Se ha utilizado **MySQL** para gestionar la BD porque además de que dispone de versiones para Windows, Linux y Mac, se trata de software libre y por tanto sería fácilmente portable a otros gestores como por ejemplo Oracle, ya que se han utilizado comandos estándar de SQL. Además, MySQL tiene una velocidad muy elevada para las consultas, y eso es un requisito fundamental para esta herramienta. También es sencillo de conectar con aplicaciones web; para muchos el mejor en este ámbito. En cuanto a la seguridad, en este proyecto no se requiere un nivel demasiado alto, por lo que MySQL es un gestor más que válido en ese sentido. Además de estos motivos, también hay uno personal, y es que en la asignatura de Bases de datos que cursé de Erasmus utilicé otro gestor: Microsoft SQL Server. Tenía curiosidad en probar otro diferente y también consideré que era bueno para mi aprendizaje personal.

La base de datos (desde ahora en adelante, BD para abreviar) se bautizó como “*erasmus_db*”, pero es un nombre que puede ser corregido en los ficheros SQL de creación de la BD y en el código de la aplicación. También se creó un usuario local administrador para que hubiera un acceso seguro a esta BD. El usuario se identificó con unas credenciales, pero estos datos también se pueden modificar. Tiene los permisos de lectura, escritura y modificación de la BD. En cuanto al contenido, la BD consta de varias tablas:

- **universities:** en esta tabla se almacenan las universidades que han formado parte en convalidaciones. Incluye los datos de la EPS, y de universidades extranjeras. Las columnas para esta tabla son las siguientes:
 - *uni_id*: en esta columna se guardan los identificadores para cada registro de universidad que se introduzca. Es un campo auto-incremental que comienza en 1 y sigue un orden ascendente para cada registro nuevo.
 - *uni_name*: incluye el nombre completo de la universidad.

- *uni_city*: almacena la ciudad en la que se encuentra la universidad.
 - *uni_country*: guarda el país de la universidad.
 - *uni_inactive*: indica si la universidad está inactiva para realizar convalidaciones de asignaturas pertenecientes a ella. Este campo sirve para que, si una universidad se borra del programa Erasmus, pero luego vuelve a estar activa simplemente se cambie este campo y así vuelva a pertenecer oficialmente a los registros activos.
- ***subjects***: sirve para almacenar todas las asignaturas con las cuales se han llevado a cabo convalidaciones, tanto de la EPS como de otras universidades. Los registros se almacenan según las siguientes columnas:
 - *subj_id*: almacena los identificadores de todas las asignaturas en orden ascendente según se introducen a la tabla *subjects*.
 - *uni_id*: guarda el identificador de la universidad a la cual pertenece cada asignatura. De esta manera relacionamos las tablas y es más sencillo realizar operaciones con ellas.
 - *subj_name*: guarda el nombre completo de las asignaturas.
 - *subj_code*: en esta columna están los códigos oficiales que identifican a cada asignatura.
 - *subj_year*: almacena e indica el año de la asignatura registrada.
 - *subjects*: guarda en formato decimal el número de créditos ECTS de los cuales consta la asignatura.
 - *subj_semester*: almacena el número de semestre en el que se realizó la asignatura (1, 2 o 0 en caso de que no se sepa).
 - *subj_inactive*: representa la inactividad de las asignaturas, al igual que en el caso de las universidades. Sin embargo, en este caso hay también dependencia de la universidad a la que pertenece, ya que si esta se encuentra inactiva la asignatura también lo estarán. Esta dependencia no se puede introducir como tal en la BD, pero se deberá manejar con el código de la aplicación.
 - ***convalidations***: es una tabla fundamental de la BD. Almacena los datos acerca de todas las convalidaciones que se han realizado, relacionando asignaturas de la EPS con asignaturas extranjeras. Los campos de esta tabla son:
 - *conv_id*: esta columna consiste en los identificadores de cada convalidación introducida en la BD, en orden ascendente según se guardaron.
 - *subj_uam_id*: guarda el identificador de las asignaturas de la EPS, UAM
 - *subj_ext_id*: es el id de las asignaturas de universidades extranjeras.
 - *conv_mixed*: este campo booleano (0 o 1) indica si una asignatura es convalidación de más de 2 asignaturas. Es decir, sería 0 si se tratase de una convalidación de 1 asignatura de la EPS por otra asignatura extranjera, y tomaría el valor 1 caso de que se hubiese convalidado 1 o más asignaturas de la EPS por más de una asignatura de otra universidad o viceversa.
 - *conv_inactive*: al igual que en las tablas anteriores, este campo indica la inactividad de una convalidación. A su vez, si una asignatura o una universidad de las incluidas en una convalidación está inactiva, la convalidación lo estará también. Igual que he comentado anteriormente, esta relación solo se manejará desde el código de la aplicación.

Field	Type	Null	Key	Default	Extra
uni_id	int(11)	NO	PRI	NULL	auto_increment
uni_name	varchar(60)	NO		NULL	
uni_city	varchar(20)	NO		NULL	
uni_country	varchar(20)	NO		NULL	
uni_inactive	tinyint(1)	NO		NULL	

Figura 14: Campos de la tabla *universities*

Field	Type	Null	Key	Default	Extra
subj_id	int(11)	NO	PRI	NULL	auto_increment
uni_id	int(11)	NO		NULL	
subj_name	varchar(80)	NO		NULL	
subj_code	varchar(30)	YES		NULL	
subj_year	varchar(7)	YES		NULL	
subj_ects	decimal(3,1)	YES		NULL	
subj_semester	tinyint(4)	YES		NULL	
subj_inactive	tinyint(1)	NO		NULL	

Figura 15: Campos de la tabla *subjects*

Field	Type	Null	Key	Default	Extra
conv_id	int(11)	NO	PRI	NULL	auto_increment
subj_uam_id	int(11)	NO		NULL	
subj_ext_id	int(11)	NO		NULL	
conv_mixed	tinyint(1)	NO		NULL	
conv_inactive	tinyint(1)	NO		NULL	

Figura 16: Campos de la tabla *convalidations*

Además de estas tres tablas principales, se crea otra cuarta y adicional denominada “*conv_initial_load*”, que se utiliza como apoyo para cargar los registros de los Excel. Sin embargo, una vez cargados los registros se pasan sus datos a “*convalidations*” y se opera con esta última y podemos eliminarla.

Para agilizar el trabajo de creación de la BD, utilicé ficheros de extensión bat que, al ser ejecutados, cargan ficheros SQL con las queries correspondientes. En el Manual de instalación, están disponibles las instrucciones y la explicación para llevar a cabo esta tarea de instalación si se desea.

Posteriormente, una vez que se crea la BD con sus tablas, hemos de introducir los registros. Los datos iniciales provienen de la lectura automatizada mediante funciones macros de VBA de los ficheros EXCEL existentes en la web de la EPS, los cuales están formados por convalidaciones de asignaturas de alumnos Erasmus del grado de Telecomunicaciones durante los cursos anteriores y el actual.

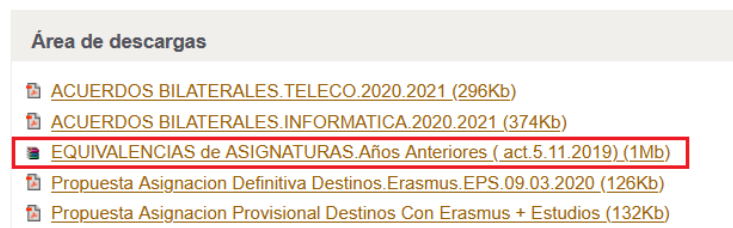


Figura 17: Localización de los ficheros Excel en la web de la EPS

A la vez que estos datos son leídos se van guardando (también gracias a VBA) en la BD. Escogí estos datos porque considero que es una fuente importante y puede considerarse el motor precursor de este trabajo. Además, contiene múltiples registros completos y oficiales, lo cual es bastante útil para probar correctamente el funcionamiento de la herramienta. Asimismo, se acordó con el tutor acerca de la realización de esta carga de datos y llegamos a la conclusión de que era algo muy a tener en cuenta para este trabajo.

Como paso final, también utilizando código SQL, hay un ajuste y normalización de valores, por ejemplo, borrados de blancos, sintaxis de fechas...

Para explicar esta sección mejor, a continuación se muestran las figuras de cómo sería el almacenamiento para algunos registros:

uni_id	uni_name	uni_city	uni_country	uni_inactive
9	POLITECNICO DI MILANO	I MILANO 02	ITALIA	0

Figura 18: Ejemplo de registro de la tabla *universities*

subj_id	uni_id	subj_name	subj_code	subj_year	subj_ects	subj_semester	subj_inactive
822	9	GEOPHYSICAL AND RADAR IMAGING	94788	2016-17	10.0	0	0
824	9	MICROWAVE ENGINEERING	94791	2016-17	5.0	0	0
827	9	POSITIONING AND TECHNOLOGIES FOR MOBILITY	96113	2016-17	5.0	0	0
890	9	ADVANCED COMPUTER ARCHITECTURES	88949	2019-20	5.0	0	0
899	9	BUSINESS INFORMATION SYSTEMS 1	91023	2019-20	5.0	0	0
900	9	BUSINESS INFORMATION SYSTEMS 2	90874	2019-20	5.0	0	0
901	9	CAMPI ELETTRROMAGNETICI	88680	2019-20	10.0	0	0
923	9	MICROCONTROLLORI	99276	2019-20	5.0	0	0
926	9	MICROWAVE ENGINEERING	94791	2019-20	5.0	0	0
927	9	MULTIMEDIA INTERNET	89043	2019-20	10.0	0	0

Figura 19: Ejemplo de algunos registros de la tabla *subjects*

conv_id	subj_uam_id	subj_ext_id	conv_mixed	conv_inactive
4	218	918	1	0
5	218	935	1	0
6	195	881	1	0
7	195	889	1	0
8	184	831	0	0
19	219	925	0	0
20	219	890	0	0
21	196	872	1	0
22	196	873	1	0
23	185	832	0	0

Figura 20: Ejemplo de algunos registros de la tabla *convalidations*

3.1.2 Aplicación web

Como se ha visto en el punto de Estado del arte, hay múltiples posibilidades para llevar a cabo el desarrollo de la aplicación web. A continuación, listaré y explicaré las tecnologías que se han utilizado.

Para la parte puramente del servidor se decidió utilizar **Java**. Aparte de los propios ficheros Java explicados más abajo, también se utiliza en otros lugares y con diversas funcionalidades estando la gran parte incluida en ficheros **JSP** (Java Server Pages).

- Se utiliza **JSP** en gran parte de los programas, tanto en la parte de Backend para interactuar con la BD, como en la de Frontend para mostrar datos como si se tratase de ficheros HTML. Se llegó a la elección de JSP al descubrir que era un lenguaje útil para esta herramienta por varios motivos en concreto:
 - Combinación de Java + HTML: JSP es un lenguaje relativamente sencillo de aprender ya que combina Java y HTML, y para esta herramienta no se requería un lenguaje específico para manejar la estructura no visible. Los archivos JSP generan directamente la parte visual al estar realizados en HTML, pero a la vez permiten acceder a la BD con su código Java. Por esta razón, JSP es un lenguaje que consta con un alto rendimiento y velocidad, sin necesidad de invocar a otros archivos extra e innecesarios. Por otro lado, Java y HTML por separado, son lenguajes con un gran respaldo a nivel comunitario y cuentan con multitud de fuentes para llevar a cabo diversos programas. Es por eso que JSP, como combinación de ambos lenguajes, no podía ser menos y existe un amplio número de programadores que lo utilizan.
 - Manejo de datos de la BD: estudiando las capacidades de JSP, descubrí que se adaptaba bien a la interacción con la BD. Además, es un lenguaje que trabaja de una manera bastante eficaz con el SGBD escogido: MySQL.
 - Posibilidad de manipulación: al tratarse de un lenguaje totalmente escrito por el desarrollador de la aplicación, esto le permite personalizarlo totalmente a su gusto tanto visual como técnicamente, sin ningún formato preestablecido en el archivo. En el caso concreto de este proyecto, esto era fundamental ya que había que tratar delicadamente la BD y la forma de mostrarla al usuario.
 - Portabilidad: JSP es una buena elección para el lenguaje de Backend Web si se busca que la aplicación sea multiplataforma. En todo momento, se ha utilizado código estándar que lo permite, y JSP no podía ser menos.

Como he mencionado, las funciones principales de JSP en la herramienta son la interacción de la aplicación web con la BD, y la muestra de los datos de la BD por pantalla. Los ficheros que utilizan JSP se organizan en los directorios que se muestran en la imagen:

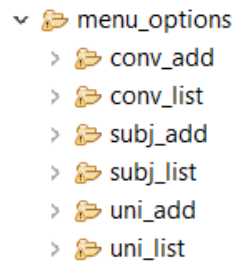


Figura 21: Organización de los ficheros JSP en la aplicación

Como se puede observar, hay dos directorios (*add* y *list*) para cada tabla principal de la BD (*convvalidations*, *subjects* y *universities*). Los directorios que acaban en *add* contienen dos ficheros JSP:

- 1) *index.jsp*: una página con un formulario que será completado por los administradores con nuevos registros para la BD.
- 2) *Data_insert.jsp*: inserta los datos obtenidos en el formulario en la BD en la tabla correspondiente.

Por otro lado, los directorios que acaban en *list* contienen 4 ficheros JSP cada uno:

- 1) *index.jsp*: página para mostrar a los administradores la información en las tablas (más adelante se explicará el manejo de estas tablas).
 - 2) *index_user.jsp*: igual a *index.jsp* pero para usuarios no administradores.
 - 3) *edit.jsp*: página de actualización de registros. Cuando un administrador quiere editar un registro llega a esta página, la cual es un formulario parecido al de la sección de *add*.
 - 4) *Data_update.jsp*: una vez que se introducen las nuevas modificaciones, es necesario actualizar la base de datos. Este fichero tiene esta función.
- También se utiliza puramente lenguaje **Java** para algunos ficheros de elevada importancia y que son puramente de la parte de Backend Web, como se ha comentado anteriormente. Entre ellos destacan:
 - 1) *DbConnection.java*: se utiliza para realizar la conexión con la BD, la cual se lleva a cabo fácilmente en pocas líneas de código gracias al controlador JDBC (también llamado JDBC driver, expuesto en el punto de Estado del arte).
 - 2) *ServletListaX.java*: se invocan desde los ficheros *list*. Recogen los datos necesarios de la BD y los envían a los ficheros *list* para que muestren la información por pantalla. Con la variable *X* se hace referencia a los distintos Servlets existentes: al haber tres tablas en la BD y dos tipos de usuarios, necesitamos en total seis Servlets. Los Servlets desarrollados se muestran debajo, en la figura 22.

- 3) *ConexionPool.java*: este fichero es invocado desde cada Servlet. Gracias a su uso se pueden manejar y reutilizar cómodamente distintas conexiones con la BD.

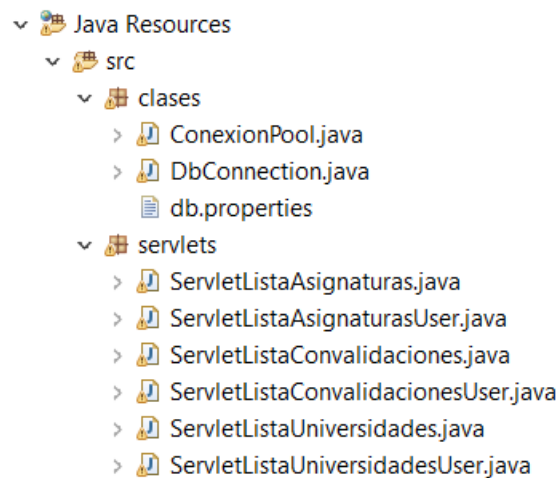


Figura 22: Ficheros java de la aplicación

En cuanto a las tecnologías del lado del cliente, se tomó como elección la combinación de **HTML5**, **CSS3** y **JavaScript**. Estudiaré el porqué de la elección de cada uno, su uso y las funciones que llevan a cabo.

- **HTML5** es un lenguaje muy útil, y que puede ser notablemente eficaz y amplio si se combina correctamente con otros lenguajes. Además, permite la ejecución offline de las páginas o aplicaciones desarrolladas gracias a él. Sin embargo, es importante no abusar de él y utilizarlo junto a otras herramientas para no estancarnos en una página web estática, y poder realizar una aplicación web. Se utiliza en la página inicial del programa, *index.html* (para usuarios) y *UAM_Admin.html* (para administradores) y para tres archivos que mantienen elementos fijos: *header.html* (cabecera con varias imágenes que contienen enlaces), *sidebar.html* (menú con las funcionalidades de la aplicación para administradores, situado al lado izquierdo de la aplicación) y *sidebar_user.html* (igual que este último pero para usuarios). Como se ha comentado en este 3.1.2, un poco más arriba, también se utiliza HTML junto a Java en todos los ficheros JSP, para utilizarlos a la vez como archivos visibles para el usuario y como archivos Backend de acceso a los registros BD.

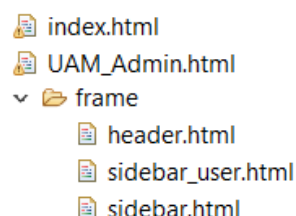


Figura 23: Archivos HTML de la aplicación

- El uso de **CSS** es prácticamente obligatorio habiendo usado HTML. Se utiliza la versión 3, la cual es más completa y funciona muy bien junto a HTML5 para dar formato a la mayoría de los objetos que ve el usuario. Todos los archivos que tienen una parte visual incluyen mediante una etiqueta al archivo *main.css*. Desde ese archivo se configura todo el formato, pudiendo referenciar de 4 formas distintas. Algunos ejemplos podrían ser: `body{ }`, `#head_static`, `.container`, `table.center`. También hay varios ficheros CSS preestablecidos provenientes en su mayoría de bibliotecas de JavaScript como jQuery, para darle formato a los elementos de JavaScript que nacen a partir de esas librerías. Esto se muestra en la siguiente captura, donde también cabe destacar el uso de imágenes existentes en la web para algunos iconos de la aplicación.

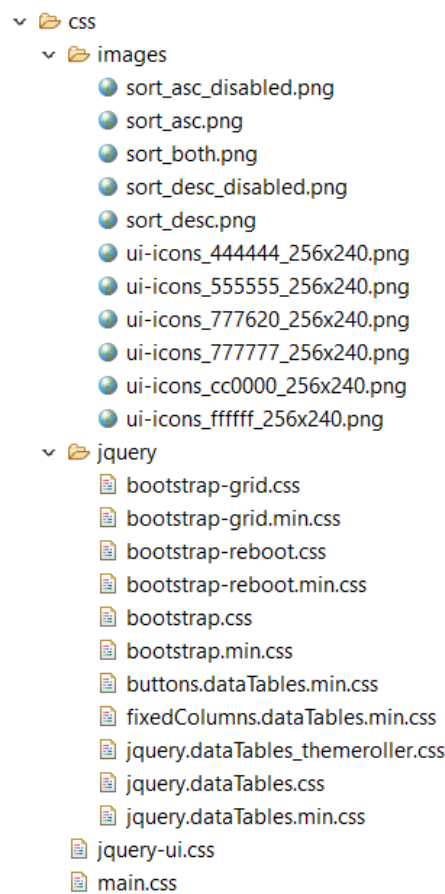


Figura 24: Organización de los ficheros CSS en la aplicación web

- **JavaScript** se ha utilizado sobre todo incluido en la parte de HTML de los archivos JSP, mediante la etiqueta `<script>`, para dar forma completa a la tabla donde se alojarán los registros que el usuario reclama, para funciones de comprobación de formularios o para cargar otros ficheros y bibliotecas. También se utiliza para el desarrollo de utilidades gráficas y efectos. Es fundamental hablar de **jQuery** (ya expuesto en el punto de Estado del arte), sin la cual no hubiera sido posible la gran mayoría de tareas de este proyecto. Gracias a sus sub-bibliotecas y plugins como son jQuery-UI, Datatables o Bootstrap, se consigue darles gran utilidad a los elementos de la aplicación, a la vez que un aspecto mejor. Además, jQuery proporciona la interacción con **AJAX** (presente también en los ficheros JSP), tecnología que comunica con el

lado del servidor mediante objetos **JSON** gracias a los Servlets de las distintas tablas y usuarios. De esta forma se accede a los registros de la BD tanto para visualizarlos, como para editarlos o añadir uno nuevo.

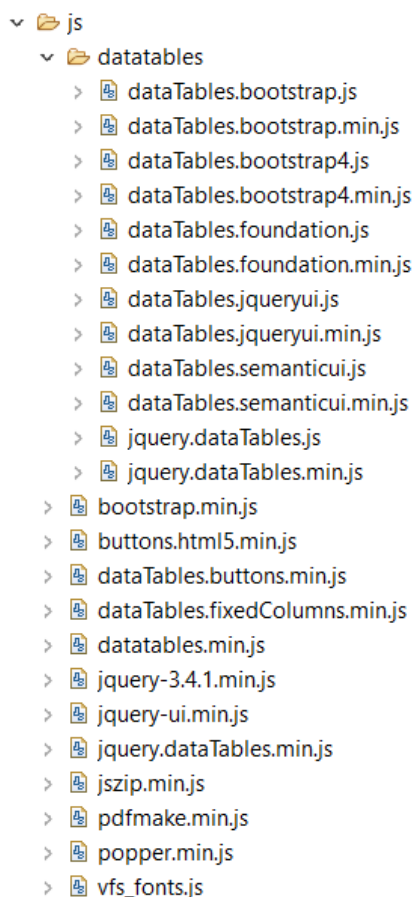


Figura 25: Organización de algunos ficheros y bibliotecas JavaScript

```
<script>
$(document).ready(function() {
    var dataTable = $('#conv_tabla').dataTable({
        scrollY: '40vh',
        scrollCollapse: true,
        paging: true,
        sScrollX: "100%",
        ajax:{
            method: "POST",
            url: "/Erasmus/ServletListaConvalidaciones",
            dataSrc: "conv_datos"
        },
        columns: [
            {"data": "uni_name"},
            {"data": "subj_year"},
            {"data": "subj_semester"},
            {"data": "subj_uam_code"},
            {"data": "subj_uam_name"},
            {"data": "subj_uam_ects"},
            {"data": "subj_ext_code"},
            {"data": "subj_ext_name"},
            {"data": "subj_ext_ects"},
            {"data": "conv_mixed"},
            {"data": "conv_inactive"},

```

Figura 26: Ejemplo de uso de JavaScript + AJAX para mostrar la tabla de convalidaciones

3.2 Esquema completo de la herramienta

Así sería el esquema que define las partes de la herramienta y la unión entre ellas:

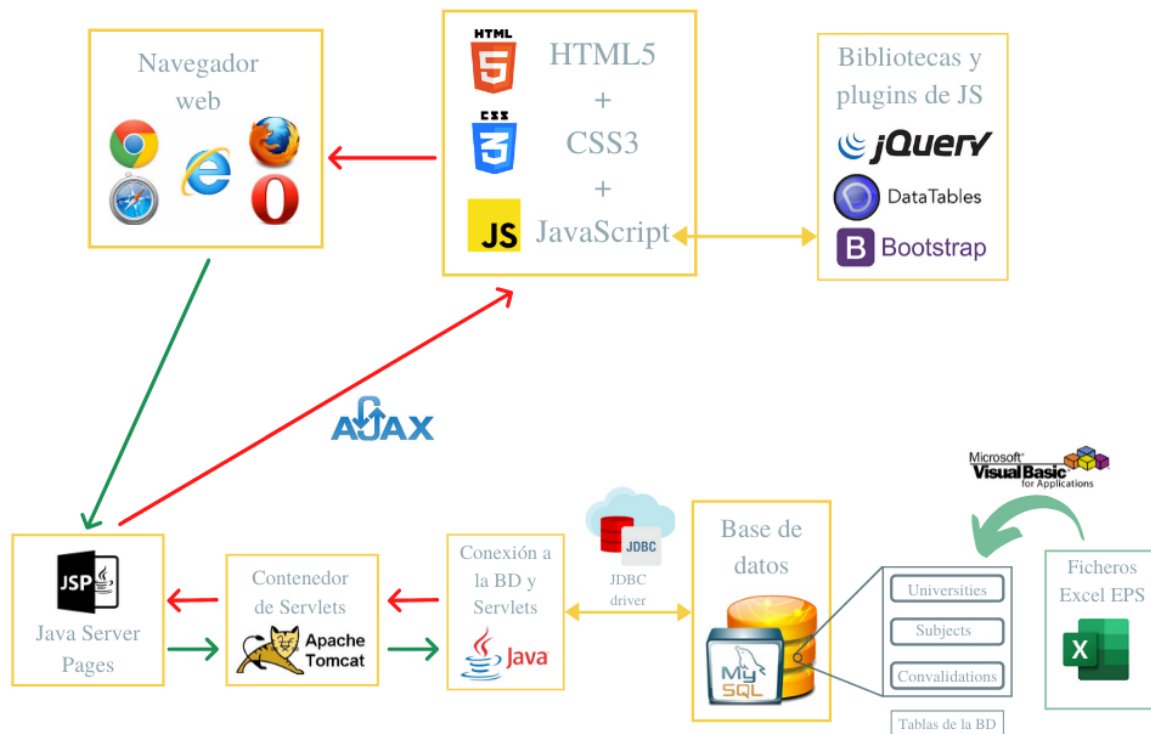


Figura 27: Esquema completo de la herramienta diseñada. Realizado gracias al sitio web Canvas

Se ha comentado prácticamente cada módulo de esta imagen por separado, pero aquí se muestra con detalle el funcionamiento de la herramienta para mostrar datos. Todo comienza con una petición del navegador, la cual es recibida y procesada por una Java Server Page. Esta envía la solicitud al Tomcat, que contiene los Servlets, y que llamará a estos para solicitarles la información que el navegador ha pedido. Estos estarán conectados a la BD y sus correspondientes tablas, las cuales contendrán los datos de los ficheros Excel de Erasmus de la web de la EPS. El Servlet obtendrá los datos y los mandará de vuelta a un fichero JSP. Este fichero enviará la información por medio de AJAX a la parte del cliente, la cual la recibirá y la entregará al navegador gracias a la combinación de HTML5, CSS3 y JavaScript, además de las bibliotecas que esta última emplea.

La ventaja principal de esta tecnología es que la sintaxis y lógica de conexión a la BD reside en el servidor y está oculta a los usuarios.

4 Desarrollo

La elaboración y la mejora de esta herramienta han seguido un camino lineal hasta su finalización. El primer paso fue estudiar las tecnologías disponibles y las posibilidades de combinación entre ellas. Una vez que se conocían todas, se escogieron las que cumplían mejor los requisitos para llevar a cabo el proyecto, y se inició el diseño de este. Se diseñó la BD, y se utilizó primero una estructura más básica para la aplicación web para manejar los datos, para finalmente acabar llegando al diseño expuesto en el punto 3 de esta memoria. En esta sección se explicarán con más detalle todas las funcionalidades de la herramienta desde una visión a la par técnica y no técnica, así como las diferencias entre los tipos de usuarios y las funcionalidades que tienen asignadas. Por último, se explicará también cómo se ha llevado a cabo el control de errores en la herramienta y el sistema de seguridad implantado para dar acceso a los datos de una forma u otra según el tipo de usuario.

4.1 Roles de usuario

Se ha realizado una aplicación web, subdividida a su vez en dos aplicaciones web: una para administradores y otra para alumnos/usuarios estándar/usuarios sin acceso directo a la BD. En esta sección nos centraremos más en las diferencias entre las capacidades de cada tipo de usuario. Sin embargo, en la subsección 4.4 Seguridad se explica más acerca de cómo se separan ambas aplicaciones para proteger los datos de la BD y solo ceder acceso a los usuarios administradores.

4.1.1 Administradores

Este grupo incluye a las personas que tienen acceso directo a la modificación y visualización de la BD desde la aplicación web. Es decir, las acciones posibles para estos usuarios son:

- Visualización de todos los registros de la BD.
- Ordenación y búsqueda de registros.
- Inserción de registros nuevos a la BD.
- Modificación de uno o varios registros de la BD.

4.1.2 Usuarios estándar (alumnos)

Por otro lado, existen los usuarios estándar o alumnos. Este tipo de usuarios no puede acceder a la BD para modificar registros existentes, ni puede tampoco insertar registros nuevos. De esta forma, sus capacidades simplemente se reducen a dos:

- Visualización de registros no inactivos de la BD: similar a la visualización para administradores, pero en este caso los usuarios estándar solo verán los registros que estén activos, es decir, habilitados por los administradores (la inactividad de registros se explicará en detalle en el 4.2.5).
- Ordenación y búsqueda de registros.

4.2 Funcionalidades de la aplicación web

En esta sección se explicará en detalle las posibles acciones a realizar desde la aplicación web, dependiendo también del tipo de usuario que las lleve a cabo.

4.2.1 Página inicial

La aplicación web comienza en una página inicial o homepage, la cual se trata de un archivo HTML5. Como se mencionó brevemente en la parte de diseño (Aplicación web), existen dos páginas de inicio según el tipo de usuario: *index.html* y *UAM_Admin.html*. Ambas poseen: una cabecera (*header.html*) con tres imágenes las cuales poseen links a sus respectivas páginas web, una parte central con una imagen de la Escuela, y un menú desplegable en la parte izquierda *sidebar.html*, desde donde se realiza la elección de la acción que se va a tomar, y desde la cual podemos regresar a esta página inicial pulsando la opción Home. La única – pero notable – diferencia es que en el menú lateral las posibilidades para administradores y usuarios estándar no son las mismas. Como se ha comentado en la sección Roles de usuario, el usuario estándar solamente es capaz de visualizar registros, mientras que los administradores tienen además de la visualización, la capacidad de insertar y/o modificar registros. A continuación, se muestra la página inicial (común) y las opciones del menú desplegable para cada tipo de usuario:



Figura 28: Página inicial de la aplicación web

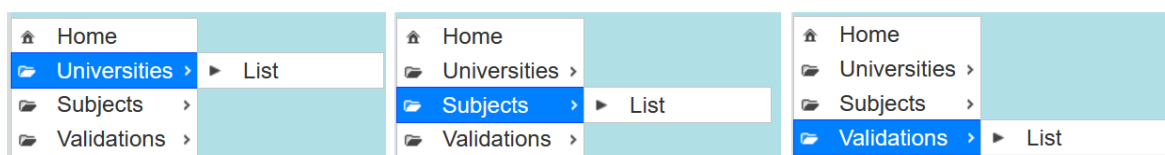


Figura 29: Menú desplegable para usuarios estándar

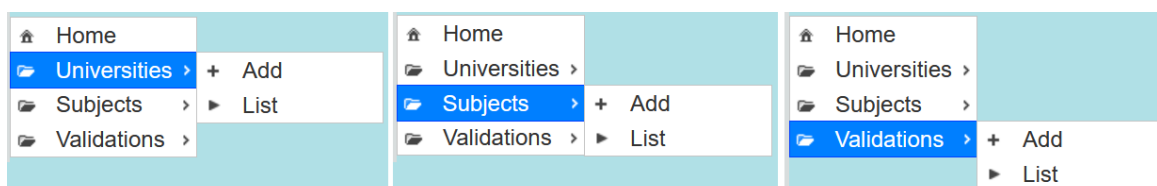


Figura 30: Menú desplegable para administradores

Se puede observar que esta página inicial es el lugar desde el que se elige un camino otro, seleccionando la opción de visualización o inserción de las distintas categorías, a gusto o necesidad del usuario que esté ejecutando la aplicación web. Como se muestra, el menú de la barra lateral izquierda es distinto para los dos tipos de usuarios.

4.2.2 Inserción de registros

Una de las opciones que se mencionaban es la inserción de nuevos registros o datos. Esta posibilidad está solo al alcance de los administradores. Al pulsar en la opción de Add que aparece en la Figura 26, se ejecuta el archivo JSP correspondiente al de la tabla en la que se quiere insertar un nuevo registro (universidades, asignaturas o convalidaciones), habiendo un formulario para cada una.

- **Universidades:** Si desea insertar una nueva universidad que ha formado parte en un programa de Erasmus, puede hacerlo desde este formulario:

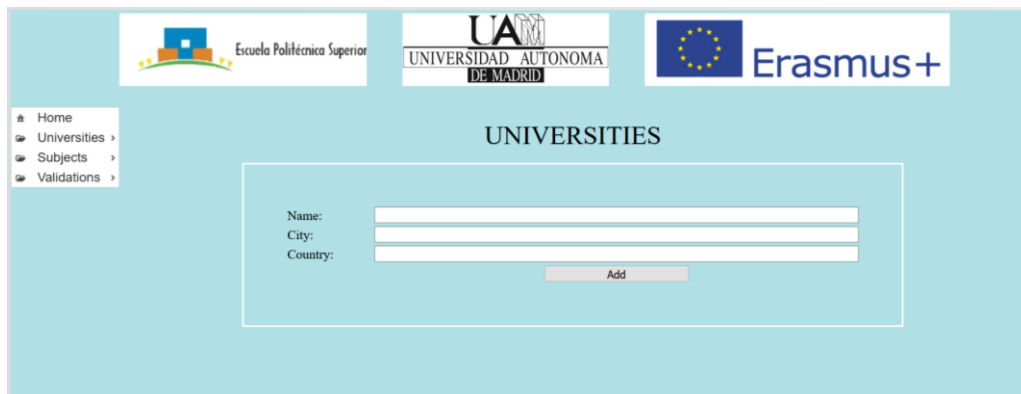


Figura 31: Formulario de la aplicación web para añadir universidades

Solo tendrá que insertar el nombre, ciudad y país de la universidad. Cuando pulse el botón Add, el nuevo registro se añadirá automáticamente a la BD. Esto ocurre gracias a la llamada al fichero *Data_insert.jsp* que accede a la BD y ejecuta la query SQL necesaria para insertar un nuevo registro, al cual le asignará un id (*uni_id*) de forma automática según el número de registro que ocupe en la tabla de *universities*.

- **Asignaturas:** Si el administrador quisiera añadir un nuevo registro de una asignatura, llegaría a este otro formulario:

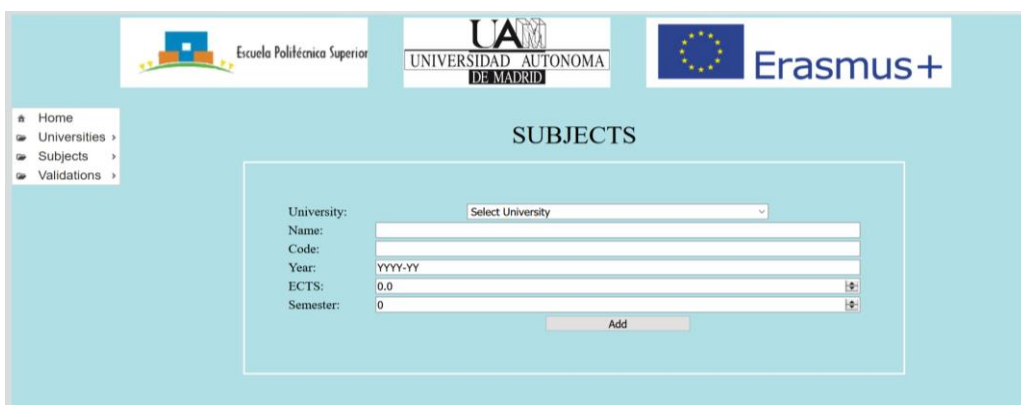


Figura 32: Formulario de la aplicación web para añadir asignaturas

Es similar al formulario de universidades, pero en este caso se deben rellenar e insertar más datos. La universidad donde se impartió la asignatura se elige a partir de una barra desplegable que muestra todas las universidades registradas en la BD. También deberá insertar el nombre de la asignatura, el código de esta, el año en que se cursó (en el formato que se indica), el número de créditos ECTS, y el semestre. Para estos dos últimos campos hay dos flechas para aumentar el número hasta la cantidad que se desee insertar. Al igual que en el caso anterior, al pulsar en el botón de Add, el fichero *Data_insert.jsp* realizará la query correspondiente para insertar el registro en la BD en la tabla *subjects*.

- **Convalidaciones:** Por último, si lo que el administrador necesita es insertar una convalidación efectuada entre dos asignaturas, bastará con que pulse en la opción Add del menú lateral (en la pestaña Validations), la cual le llevará a otro formulario parecido a los anteriores:

Figura 33: Formulario de la aplicación web para añadir convalidaciones

Sin embargo, en este caso se rellenan tres campos, todos mediante barras desplegables. El primero contiene las asignaturas que pertenecen a la EPS existentes en la BD; el segundo son las asignaturas de universidades extranjeras también de la BD; y el tercero es una elección entre Yes/No, para indicar si la convalidación es mixta (involucra más de dos asignaturas) o no. Es decir, cuando el administrador vaya a insertar una convalidación, deberá seleccionar la asignatura de la UAM y la asignatura extranjera por la cual se convalidó, e indicar si la convalidación es mixta o no. Si se tratase de una convalidación mixta (más de 2 asignaturas) simplemente tendrá que marcar la opción Yes en la tercera barra desplegable, e ir insertando las asignaturas con convalidación en formato “1:1”, cambiando las asignaturas que correspondan.

Se muestra un ejemplo para entenderlo mejor: un administrador quiere insertar la siguiente convalidación en la BD.

REDES MULTIMEDIA	COMPUTER NETWORKS
	COMPUTER NETWORKS SECURITY

Tabla 1: Ejemplo de convalidación mixta o con asignaturas múltiples

El administrador llegará al formulario, indicará Yes en el tercer desplegable al tratarse de una convalidación mixta de tipo n:n, en este caso 1:2 en concreto. Insertará REDES MULTIMEDIA en el primer desplegable, COMPUTER NETWORKS en el segundo y pulsará Add. A continuación, el formulario se vaciará y tendrá que volver a seleccionar Yes en el tercer desplegable y REDES MULTIMEDIA en el primero. En el segundo, en

cambio, ahora escogerá COMPUTER NETWORKS SECURITY. Finalmente pulsará Add de nuevo, y así se completará la inserción de la convalidación en el formulario. En cuanto a los propios datos, viajarán hasta la BD de nuevo gracias al archivo *Data_insert.jsp*, que insertará el/los registro/s mediante una query que inserta el id de las asignaturas en lugar del nombre completo, para ahorrar procesamiento.

También es oportuno comentar que cuando se completa la inserción de los datos en la BD, aparece un mensaje de confirmación encima del formulario, con el texto “Succesfully Added”. Entonces, el formulario estará vacío y listo para una nueva inserción. A continuación, se muestra un ejemplo de una inserción correcta en el formulario de asignaturas:



The screenshot shows a web interface for adding subjects. At the top, there are logos for 'Escuela Politécnica Superior', 'UNIVERSIDAD AUTONOMA DE MADRID', and 'Erasmus+'. On the left, a navigation menu includes 'Home', 'Universities >', 'Subjects >', and 'Validations >'. The main heading is 'SUBJECTS'. Below it is a form with the following fields: 'University:' (dropdown menu showing 'AGH UNIVERSITY'), 'Name:' (text input with 'COMPUTER TECHNIQUES FOR SYGNAL PROCESSING'), 'Code:' (text input with 'int-course_031'), 'Year:' (text input with '2019-20'), 'ECTS:' (spin box with '4.0'), and 'Semester:' (spin box with '1'). An 'Add' button is at the bottom right of the form.

Figura 34: Ejemplo de inserción correcta de datos (I)



This screenshot shows the same 'SUBJECTS' form after a successful submission. A red message 'Succesfully Added: COMPUTER TECHNIQUES FOR SYGNAL PROCESSING' is displayed above the form. The form fields are now empty or reset: 'University:' (dropdown menu showing 'Select University'), 'Name:' (empty text input), 'Code:' (empty text input), 'Year:' (text input with 'YYYY-YY'), 'ECTS:' (spin box with '0.0'), and 'Semester:' (spin box with '0'). The 'Add' button remains at the bottom right.

Figura 35: Ejemplo de inserción correcta de datos (II)

4.2.3 Visualización de registros

Otra de las características fundamentales en esta aplicación web es la de la visualización de registros. Al pulsar en la barra lateral sobre la opción List de las Figuras 25 y 26, ejecutaremos respectivamente los archivos *index_user.jsp* y *index.jsp* correspondientes a la tabla que se quiere visualizar. Entonces se mostrarán los datos que se han solicitado en tablas ordenadas con el plugin DataTables de jQuery (JavaScript), que gracias a la tecnología AJAX, obtiene los datos solicitados en objetos JSON a partir del Servlet del tipo de usuario y tabla que se desea visualizar. Este mecanismo está explicado en la parte de Diseño, concretamente en la parte final de JavaScript de la subsección Aplicación web.

De forma análoga a lo comentado anteriormente, la visualización es distinta para según qué tipo de usuario la esté llevando a cabo, aunque en este caso los registros visualizados son prácticamente idénticos excepto por el campo Inactive y Edit.

El campo Inactive solo es visible para administradores. Para los usuarios, en vez de mostrarse el campo Inactive en un registro, este se mostrará o dejará de mostrar según el valor que el campo tenga. Este campo tomará un valor entre Yes o No debido a las columnas de la BD *uni_inactive*, *subj_inactive* y *conv_inactive* (explicadas en detalle en Base de datos). La inactividad de los registros se narra con más detenimiento en el apartado 4.2.5 Registros inactivos.

Además los administradores disponen de una última columna con la opción de editar cada registro. La edición de registros se explicará en el apartado 4.2.4.

Para explicar de una manera más concisa la visualización, así como algunas características extra que esta permite, se muestran algunos ejemplos:



Name	Code	Country
AALTO UNIVERSITY	SF ESPOO 12	FINLANDIA
AGH UNIVERSITY	PL KRAKOW 02	POLONIA
DANMARKS TEKNISKE UNIVERSITET	DK LYNGBY01	DINAMARCA
ECOLE D'INGENIERIE	CH DELEMONTO1	SUIZA
EPS UNIVERSIDAD AUTONOMA DE MADRID	SP MADRID 01	ESPAÑA
LINNEAUS UNIVERSITET	S VAXJO 03	SUECIA

Figura 36: Visualización de universidades en la aplicación web para usuarios estándar





[Home](#)
[Universities >](#)
[Subjects >](#)
[Validations >](#)

[Show all rows](#)
[Copy](#)
[Excel](#)
[CSV](#)
[PDF](#)

Search:

University	Code	Name	Period	ECTS	Score
AGH UNIVERSITY	UBPIO-249	CYBERSECURITY SCHEMES IN EUROPEAN UNION COUNTRIES	2017-18	4.0	0
AGH UNIVERSITY	UBPIO-225	OPERATIONS RESEARCH	2018-19	5.0	0
AGH UNIVERSITY	UBPIO-214	OPTICAL FIBERS- TECHNOLOGY AND APPLICATIONS	2017-18	4.0	0
AGH UNIVERSITY	UBPIO-133	MARKETING	2017-18	5.0	0
AGH UNIVERSITY	UBPIO-132	MANAGEMENT AND ORGANIZATION	2017-18	5.0	0
AGH UNIVERSITY	UBPIO-131	PRINCIPLES OF AUDIO ENGINEERING	2018-19	8.0	0

Showing 1 to 921 of 921 entries

Previous **1** Next

Figura 37: Visualización de asignaturas en la aplicación web para usuarios estándar





[Home](#)
[Universities >](#)
[Subjects >](#)
[Validations >](#)

[Show 10 rows](#)
[Copy](#)
[Excel](#)
[CSV](#)
[PDF](#)




Search:

University	Period	Score	UBIO Code	UBIO Name	UBIO ECTS	UBIO Code	UBIO Name	UBIO ECTS	Status
AALTO UNIVERSITY	2014-15	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0	S-96,3415	ANTENAS THEORY	5.0	Yes
AALTO UNIVERSITY	2014-15	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0	S-26,3392	ELECTROMAGNETIC COMP.	4.0	Yes
AALTO UNIVERSITY	2014-15	0	18486	ELECTRÓNICA DE COMUNICACIONES	6.0	S-69,3114	INTEGRATED RF-CIRCUIT	5.0	
AALTO UNIVERSITY	2013-14	0	15332	INSTRUMENTACIÓN ELECTRÓNICA	6.0	S-69,3114	MICROSENSORS	5.0	

Showing 1 to 10 of 515 entries

Previous **1** 2 3 4 5 ... 52 Next

Figura 38: Visualización de convalidaciones en la aplicación web para usuarios estándar

[Home](#)
[Universities >](#)
[Subjects >](#)
[Validations >](#)

[Show 10 rows](#)
[Copy](#)
[Excel](#)
[CSV](#)
[PDF](#)

Search:

University	Period	Score	UBIO Code	UBIO Name	UBIO ECTS	UBIO Code	UBIO Name	UBIO ECTS	Status	Subject	Action
AALTO UNIVERSITY	2014-15	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0	S-96,3415	ANTENAS THEORY	5.0	Yes		Edit
AALTO UNIVERSITY	2014-15	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0	S-26,3392	ELECTROMAGNETIC COMP.	4.0	Yes		Edit
AALTO UNIVERSITY	2014-15	0	18486	ELECTRÓNICA DE COMUNICACIONES	6.0	S-69,3114	INTEGRATED RF-CIRCUIT	5.0			Edit
AALTO UNIVERSITY	2013-14	0	15332	INSTRUMENTACIÓN ELECTRÓNICA	6.0	S-69,3114	MICROSENSORS	5.0			Edit

Showing 1 to 10 of 515 entries

Previous **1** 2 3 4 5 ... 52 Next

Figura 39: Visualización de convalidaciones en la aplicación web para administradores

Los datos se presentan ordenados en tablas con varias columnas, dependiendo de la categoría que estemos mostrando. Como se comenta antes de las Figuras, y como se puede observar, la diferencia principal entre las columnas de administradores y usuarios es que para los primeros está disponible la visualización de las columnas Inactive y Edit, mientras que para usuarios no.

Como funcionalidades adicionales y comunes a ambos existen las siguientes:

- Elegir cuántas filas se muestran (10, 25, 50 o todas). Si elegimos todas no hará falta pasar a la siguiente “página” de la tabla, pero es una opción que muchas webs incluyen y que puede ser útil para algunas personas.

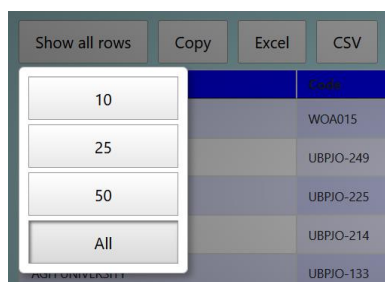


Figura 40: Posibilidad de elegir distinto número de filas a mostrar

- Ordenar los registros ascendente o descendente según una columna, simplemente pinchando sobre ella.
- Buscar uno o varios registros determinados sobre el buscador en la parte superior derecha encima de la tabla. Se pueden realizar búsqueda de registros por cualquier campo, sea número, letras o combinación de ambas. De esta forma, por ejemplo, un alumno que busque un destino Erasmus puede buscar una asignatura determinada en una universidad concreta y guiarse a partir de las convalidaciones de años anteriores. Aquí se muestran dos ejemplos de búsqueda:

Show 10 rows	Copy	Excel	CSV	PDF	Search: circuitos analógicos				
University	Period	Sem	UAM Code	UAM Name	UAM ECTS	EXT Code	EXT Name	EXT ECTS	Mixed
POLITECHNIKA WROCLAWSKA	2018-19	0	18478	CIRCUITOS ANALÓGICOS Y DE POTENCIA	6.0	ECEA00009W	ELECTRONIC CIRCUITS	7.0	
POLITECNICO DI MILANO	2015-16	0	18478	CIRCUITOS ANALÓGICOS Y DE POTENCIA	6.0	97470	POWER ELECTRONICS AND SUPPLIES	5.0	

Figura 41: Ejemplo de búsqueda de "circuitos analógicos" en la tabla de convalidaciones

Show 10 rows	Copy	Excel	CSV	PDF	Search: 124				
University	Period	Sem	UAM Code	UAM Name	UAM ECTS	EXT Code	EXT Name	EXT ECTS	Mixed
DANMARKS TEKNISKE UNIVERSITET	2018-19	0	18488	Ingeniería Acústica	6.0	31241	Building Acoustics	5.0	Yes
DANMARKS TEKNISKE UNIVERSITET	2015-16	0	18488	INGENIERÍA ACÚSTICA	6.0	31241	BUILDING ACOUSTICS	5.0	

Figura 42: Ejemplo de búsqueda de "124" en la tabla de convalidaciones

Show all rows	Copy	Excel	CSV	PDF	Search: antenas pisa				
University	Period	Sem	UAM Code	UAM Name	UAM ECTS	EXT Code	EXT Name	EXT ECTS	Mixed
UNIVERSITÀ DEGLI STUDI DI PISA	2017-18	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0	283II	ANTENNE E PROPAGAZIONE	9.0	
UNIVERSITÀ DEGLI STUDI DI PISA	2016-17	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0		ANTENNE E PROPAGAZIONE	6.0	
UNIVERSITÀ DEGLI STUDI DI PISA	2015-16	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0		ANTENNE E PROPAGAZIONE	9.0	

Figura 43: Ejemplo de búsqueda de "antenas pisa" en la tabla de convalidaciones

- Copiar los registros visibles al portapapeles: una vez se selecciona el número de filas a mostrar y se ordena la tabla de una forma determinada, el usuario puede copiar directamente a su portapapeles los datos, y hacer con ellos lo que desee. Además, el usuario puede realizar una búsqueda y copiar los registros que aparezcan como resultado. De esta manera puede elaborar poco a poco un archivo personalizado con las universidades/convalidaciones que le interesen.

Show all rows	Copy	Excel	CSV	PDF	Search: antenas pisa				
University	Period	Sem	UAM Code	UAM Name	UAM ECTS	EXT Code	EXT Name	EXT ECTS	Mixed
UNIVERSITÀ DEGLI STUDI DI PISA	2017-18	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0	283II	ANTENNE E PROPAGAZIONE	9.0	
UNIVERSITÀ DEGLI STUDI DI PISA	2016-17	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0		ANTENNE E PROPAGAZIONE	6.0	
UNIVERSITÀ DEGLI STUDI DI PISA	2015-16	0	18495	ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA	6.0		ANTENNE E PROPAGAZIONE	9.0	

Showing 1 to 3 of 3 entries (filtered from 515 total entries) Previous 1 Next

Figura 44: Copia de 3 registros al portapapeles

- Generar archivos externos a partir de la información de la tabla (PDFs, Excel, CSV): de la misma forma que el usuario copia las filas visibles al portapapeles, también puede exportar ese contenido a 3 tipos de archivos, cada uno con sus características, ventajas y desventajas:
 - PDFs: se muestra la información de una manera más limpia pero no se puede modificar.
 - CSV: la información es complicada de leer, pero es un archivo que ocupa muy poco espacio, y que puede ser útil para algún usuario.
 - Excel: se muestra la información de una manera muy ordenada y además se puede editar, pero ocupa más espacio que las dos anteriores.

Véanse las siguientes figuras de comparativa del espacio y formato de cada tipo de archivo, respectivamente, para comprender mejor lo ya expuesto.


 Validations.pdf	Adobe Acrobat D...	14 KB
 Validations.csv	Archivo de valores...	1 KB
 Validations.xlsx	Hoja de cálculo de...	16 KB

Figura 45: Comparativa del espacio que ocupa cada tipo de archivo

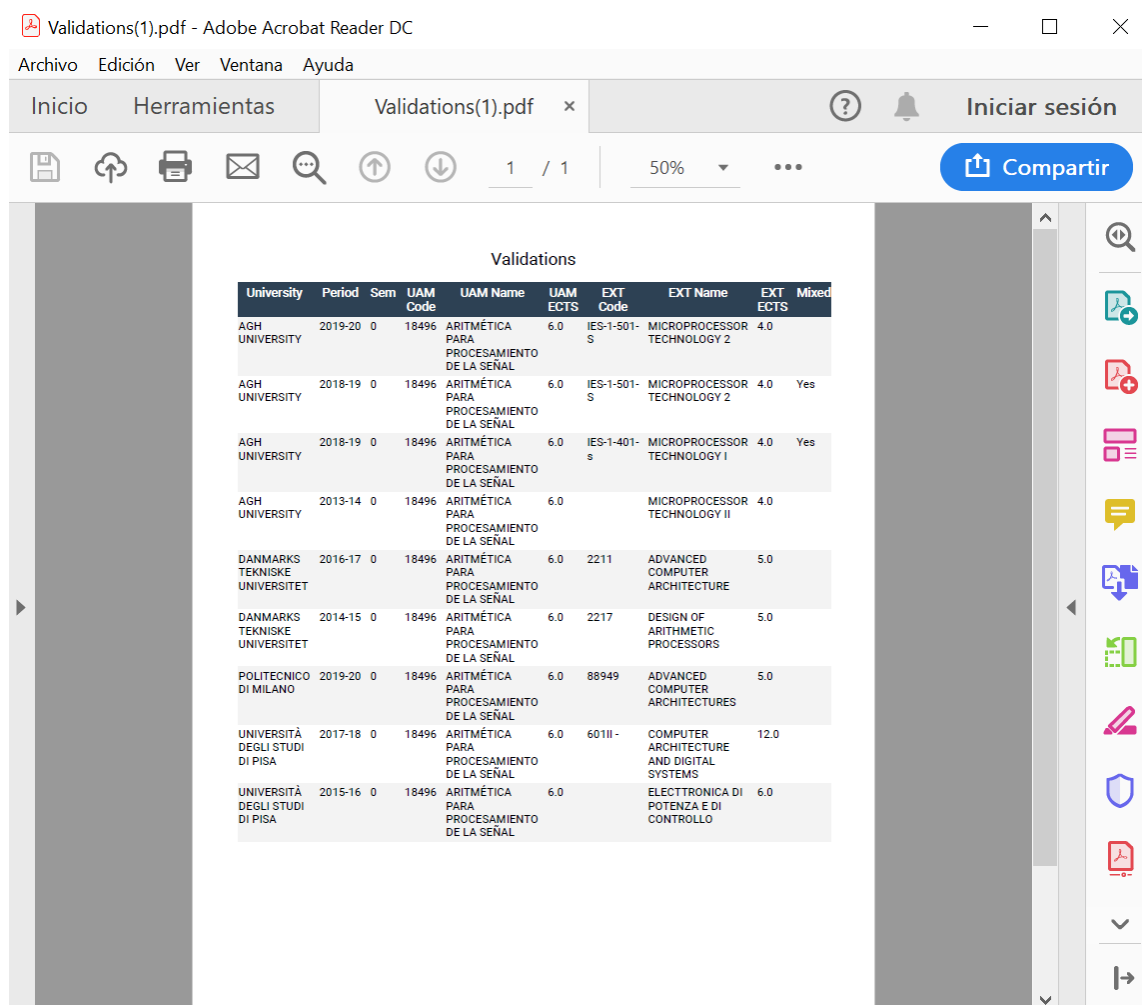


Figura 46: PDF generado con la búsqueda de "aritmética para procesamiento de la señal"

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	University,Period,Sem,UAM Code,UAM Name,UAM ECTS,EXT Code,EXT Name,EXT ECTS,Mixed													
2	AGH UNIVERSITY,"2019-20","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","IES-1-501-S","MICROPROCESSOR TECHNOLOGY 2","4.0",""													
3	AGH UNIVERSITY,"2018-19","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","IES-1-501-S","MICROPROCESSOR TECHNOLOGY 2","4.0","Yes"													
4	AGH UNIVERSITY,"2018-19","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","IES-1-401-s","MICROPROCESSOR TECHNOLOGY I","4.0","Yes"													
5	AGH UNIVERSITY,"2013-14","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","","MICROPROCESSOR TECHNOLOGY II","4.0",""													
6	DANMARKS TEKNISKE UNIVERSITET,"2016-17","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","2211","ADVANCED COMPUTER ARCHITECTURE","5.0",""													
7	DANMARKS TEKNISKE UNIVERSITET,"2014-15","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","2217","DESIGN OF ARITHMETIC PROCESSORS","5.0",""													
8	POLITECNICO DI MILANO,"2019-20","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","88949","ADVANCED COMPUTER ARCHITECTURES","5.0",""													
9	UNIVERSITÀ DEGLI STUDI DI PISA,"2017-18","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","601II -","COMPUTER ARCHITECTURE AND DIGITAL SYSTEMS","12.0",""													
10	UNIVERSITÀ DEGLI STUDI DI PISA,"2015-16","0","18496","ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL","6.0","","ELETTRONICA DI POTENZA E DI CONTROLLO","6.0",""													
11														

Figura 47: CSV generado con la búsqueda de "aritmética para procesamiento de la señal"

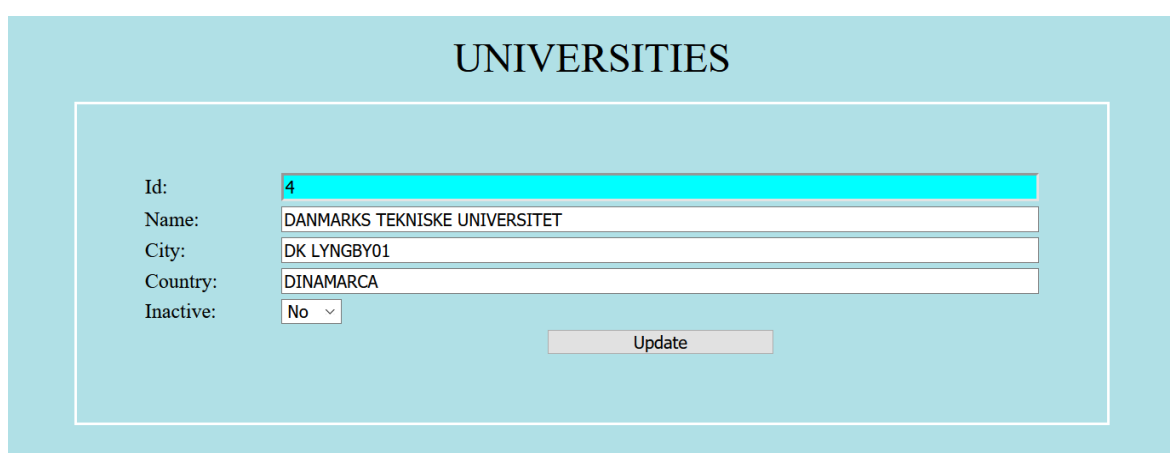
Validations										
University	Period	Sem	UAM Code	UAM Name	UAM ECTS	EXT Code	EXT Name	EXT ECTS	Mixed	
AGH UNIVERSITY	2019-20	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6	IES-1-501-S	MICROPROCESSOR TECHNOLOGY 2	4		
AGH UNIVERSITY	2018-19	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6	IES-1-501-S	MICROPROCESSOR TECHNOLOGY 2	4	Yes	
AGH UNIVERSITY	2018-19	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6	IES-1-401-s	MICROPROCESSOR TECHNOLOGY I	4	Yes	
AGH UNIVERSITY	2013-14	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6		MICROPROCESSOR TECHNOLOGY II	4		
DANMARKS TEKNISKE UNIVERSITET	2016-17	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6	2211	ADVANCED COMPUTER ARCHITECTURE	5		
DANMARKS TEKNISKE UNIVERSITET	2014-15	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6	2217	DESIGN OF ARITHMETIC PROCESSORS	5		
POLITECNICO DI MILANO	2019-20	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6	88949	ADVANCED COMPUTER ARCHITECTURES	5		
UNIVERSITÀ DEGLI STUDI DI PISA	2017-18	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6	601II -	COMPUTER ARCHITECTURE AND DIGITAL SYSTEMS	12		
UNIVERSITÀ DEGLI STUDI DI PISA	2015-16	0	18496	ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL	6		ELETTRONICA DI POTENZA E DI CONTROLLO	6		

Figura 48: Excel generado con la búsqueda de "aritmética para procesamiento de la señal"

4.2.4 Modificación de registros

La posibilidad de edición o modificación de registros es una de las capacidades de los administradores. Gracias a ella se pueden cambiar los valores de un registro y/o corregir datos incompletos o erróneos. Los administradores pueden acceder a la modificación de registros desde las tablas de visualización: cada registro posee en su última columna (Actions) un hyperlink con la palabra Edit. Este enlace ejecutará el archivo *edit.jsp* correspondiente, el cual es un archivo JSP parecido a cualquier *index.jsp* para añadir registros este proyecto. En este caso se invoca a *Data_update.jsp*, el cual realiza las queries de actualización adecuadas para el cambio realizado.

La única diferencia es que habrá campos fijos de solo lectura, que no se podrán editar. El porqué de esto se explica a continuación con algunos ejemplos:



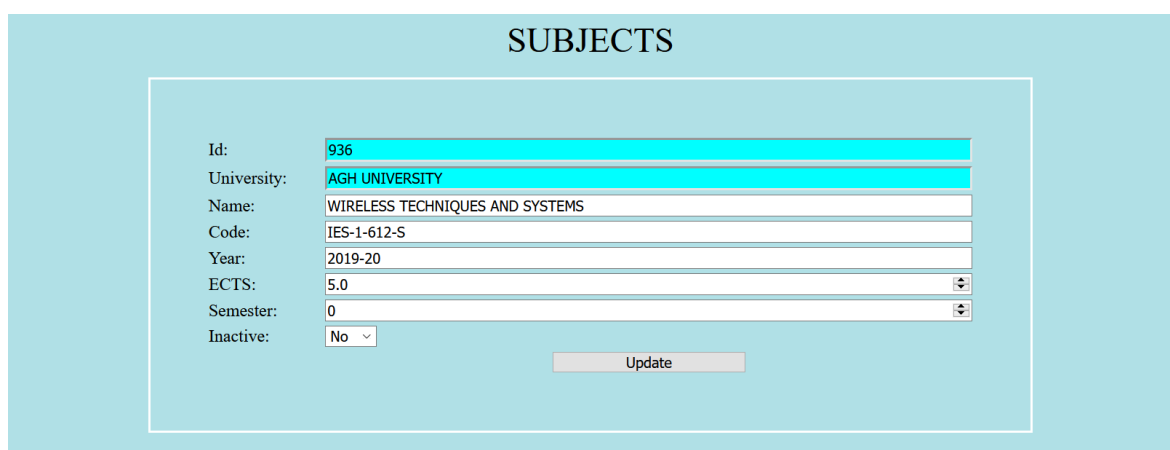
UNIVERSITIES

Id:	4
Name:	DANMARKS TEKNISKE UNIVERSITET
City:	DK LYNGBY01
Country:	DINAMARCA
Inactive:	No

Update

Figura 49: Página de edición de un registro de universidades

En este caso, el único campo no editable es el del Id, ya que es un campo fijo en la BD y que no podemos tocar a no ser que el registro sea borrado directamente desde un gestor de BD. El resto de los parámetros pueden ser modificados sin problema. Es de gran importancia el campo Inactive, cuya influencia se explica en la próxima subsección.



SUBJECTS

Id:	936
University:	AGH UNIVERSITY
Name:	WIRELESS TECHNIQUES AND SYSTEMS
Code:	IES-1-612-S
Year:	2019-20
ECTS:	5.0
Semester:	0
Inactive:	No

Update

Figura 50: Página de edición de un registro de asignaturas

Para la modificación de asignaturas, solo podremos editar los propios campos de una asignatura, ya que la universidad a la que pertenece es otro registro del cual depende, y no al revés, por lo que no podrá ser modificado.

VALIDATIONS

Validation Id:	212
Period:	2011-12
UAM University:	EPS UNIVERSIDAD AUTONOMA DE MADRID
UAM Subject:	PROYECTOS
Ext University:	AALTO UNIVERSITY
Ext Subject:	SOFTWARE PROJECT MANAGEMENT
Mixed:	Yes
Inactive:	No

Figura 51: Página de edición de un registro de convalidaciones

En tercer lugar, las convalidaciones son un tipo de registro que depende de las asignaturas que las conforman, las cuales dependen de las universidades. Es por ello que no se puede editar ninguno de esos dos tipos de registros, y solo se podrá modificar la inactividad de la convalidación, así como el campo que indica si es mixta o no.

Por último, hay que comentar que cuando se realiza una modificación correctamente aparece un mensaje parecido a cuando se inserta, que muestra “Successfully updated”.

VALIDATIONS

Successfully Updated

Validation Id:	212
Period:	2011-12

Figura 52: Mensaje de modificación correcta

4.2.5 Registros inactivos

A lo largo de esta memoria se ha hecho referencia más de una vez al concepto de la inactividad de registros. El significado de esto es muy simple: a la hora de modificar la BD, en vez de borrar un registro cualquiera, el administrador lo puede marcar como inactivo. Esto indicará si dicho registro está o no habilitado en ese momento, e influirá en los datos que se muestran al usuario. Es decir, si el campo de inactividad de un registro está activado, a los usuarios no se les mostrará ese registro cuando soliciten la visualización de una tabla. Sin embargo, los administradores podrán seguir viéndolo, aunque con la marca de inactividad (en el campo Inactive de la tabla aparecerá Yes).

Por defecto están todos los registros activos, ya que son los datos oficiales de la UAM, y no soy conocedor de si hay universidades, asignaturas o convalidaciones que no estén habilitadas para Erasmus en este momento.

Además de esto es importante comentar que, al depender todas las tablas directamente de la tabla *universities*, cuando un registro de esta se marque como inactivo, las asignaturas y convalidaciones que pertenezcan a esa universidad no se mostrarán al usuario. Lo mismo ocurre si se marca como inactivo un registro de la tabla *subjects*: las convalidaciones que contengan esas asignaturas tampoco se cargarán.

El motivo principal de realizar esta funcionalidad de esta forma, en vez de, por ejemplo, permitir el borrado directo de registros, es que se puede tener un histórico de lo realizado. Si una universidad, asignatura o convalidación borrada volviese a estar activa o habilitada, sería suficiente con restaurar el estado del registro. Por ejemplo, en el caso de inactivar un registro por error, restaurarlo sería muy sencillo.

4.3 Control de errores

Junto con las funcionalidades que la aplicación web otorga, es hora de comentar algún aspecto más que la herramienta en conjunto contiene. Uno de ellos es el control de errores que se ha llevado a cabo en el desarrollo de este proyecto.

En cuanto a la BD, al realizar la carga de los archivos Excel, se observó que algunos de ellos contenían errores pequeños de ortografía y/o formato. Es algo normal, ya que estos ficheros contienen datos que han sido escritos manualmente, y la carga de trabajo es importante, con lo que es fácil equivocarse. Es por ello que, en este proyecto, a la hora de cargar los registros a las tablas de la BD, se desarrollaron algunas queries para corregir la mayoría de estos errores. En la Figura 49 se pueden visualizar algunos de ellos. Hay que comentar que no se corrigieron todos ya que algunos eran complicados y se necesitaría la ayuda del personal de la universidad. Por ejemplo, campos vacíos o asignaturas que figuran con varios nombres o códigos. Sin embargo, dado que la BD contiene muchos registros y que la herramienta es solo una ayuda no oficial para los usuarios, no son errores que afecten realmente a gran escala a su funcionamiento.

Aparte de estos errores, no había otros preestablecidos. Los que hubo que controlar después giraron en torno a la aplicación web, siendo las comprobaciones en los formularios las principales.

Cuando un administrador está rellenando un formulario para insertar un registro cualquiera, el programa de la aplicación web comprueba, gracias a JavaScript, que no se dejen campos vacíos, y gracias a HTML, que los que se han introducido tengan el formato que se requiere. En algunos campos donde se puede aumentar el número gracias a flechas situadas a la derecha también se comprueba que el número esté dentro del rango. Como es natural, no se puede comprobar cualquier inserción que realice un administrador, ya que puede haber infinitas combinaciones y este proyecto no busca ese objetivo.

En las siguientes dos páginas se muestran las principales correcciones y el control de errores de la herramienta.

```

-- Eliminacion de blancos antes y despues
UPDATE conv_initial_load
SET conv_subj1_name = TRIM(conv_subj1_name),
    conv_subj2_name = TRIM(conv_subj2_name);
commit;

-- Normalizacion decimales con '.'
UPDATE conv_initial_load
SET conv_subj1_ects = REPLACE(conv_subj1_ects, ',', '.'),
    conv_subj2_ects = REPLACE(conv_subj2_ects, ',', '.');
commit;

-- Normalizacion creditos
UPDATE conv_initial_load
SET conv_subj1_ects = REPLACE(conv_subj1_ects, '.0', ''),
    conv_subj2_ects = REPLACE(conv_subj2_ects, '.0', '');
commit;

-- ERROR 1366 (HY000): Incorrect DECIMAL value: '0' for column '' at row -1
update conv_initial_load set conv_subj1_ects ="0" where conv_subj1_ects = "";
commit;

update conv_initial_load set conv_subj2_ects ="0" where conv_subj2_ects = "";
commit;

-- Normalizacion a YYYY-YY
UPDATE conv_initial_load
SET conv_year = REPLACE(conv_year, '-20', '-')
where conv_year not like '%19-20';
commit;

-- Correccion de la hoja excel
update conv_initial_load
set conv_subj2_name ="ANTENAS Y COMPATIBILIDAD ELECTROMAGNÉTICA", conv_subj2_code="18495", conv_subj2_ects="6"
where conv_subj2_name like "ANTENAS%";

update conv_initial_load
set conv_subj2_name ="ARITMÉTICA PARA PROCESAMIENTO DE LA SEÑAL", conv_subj2_code="18496", conv_subj2_ects="6"
where conv_subj2_name like "ARITM%";

update conv_initial_load
set conv_subj2_name ="ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES", conv_subj2_code="18496", conv_subj2_ects="6"
where conv_subj2_name like "ARQUITECTURA Y TECN%";

update conv_initial_load
set conv_subj2_name ="CRÉDITOS OPTATIVOS", conv_subj2_code="0", conv_subj2_ects="6"
where conv_subj2_name like "CR%";

update conv_initial_load
set conv_subj2_name ="ELECTRÓNICA DE COMUNICACIONES", conv_subj2_code="18486", conv_subj2_ects="6"
where conv_subj2_name like "ELECTRÓNICA DE %";

update conv_initial_load
set conv_subj2_name ="PROYECTOS Y SISTEMAS DE TELECOMUNICACIÓN", conv_subj2_code="18491", conv_subj2_ects="6"
where conv_subj2_code like "18491%";
commit;

-- Eliminacion de duplicados
CREATE TABLE conv_initial_load_2 AS select distinct * from conv_initial_load;
commit;
drop table conv_initial_load;
commit;
RENAME TABLE conv_initial_load_2 TO conv_initial_load;
commit;

```

Figura 53: Corrección de errores de la BD

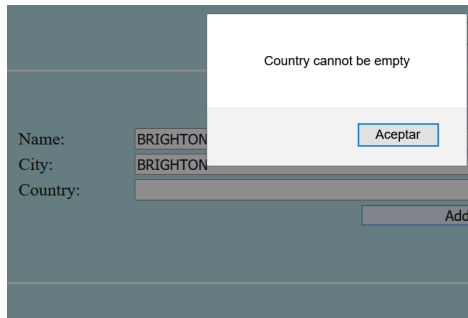


Figura 55: Comprobación de campo vacío (II)

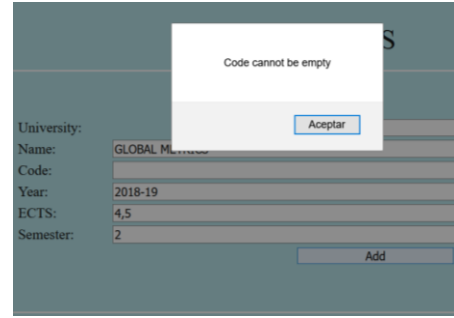


Figura 54: Comprobación de campo vacío (I)

SUBJECTS

University:	AGH UNIVERSITY
Name:	GLOBAL METRICS
Code:	agh12
Year:	2018-19
ECTS:	4,5
Semester:	3

Add

Seleccione un valor que no sea mayor de 2.

Figura 56: Comprobación de errores de rango en formulario

```
<script>
function checkempty(form) {
    if (form.uni_id.value == null ||
        form.uni_id.value == undefined ||
        form.uni_id.value < 1 ||
        form.uni_id.value.length == 0){
        alert("University cannot be empty\n");
        return false;
    }else if (form.subj_name.value == null ||
        form.subj_name.value == undefined ||
        form.subj_name.value.length == 0){
        alert("Name cannot be empty\n");
        return false;
    }else if (form.subj_code.value == null ||
        form.subj_code.value == undefined ||
        form.subj_code.value.length == 0){
        alert("Code cannot be empty\n");
        return false;
    }else if (form.subj_year.value == null ||
        form.subj_year.value == undefined ||
        form.subj_year.value.length == 0){
        alert("Year cannot be empty\n");
        return false;
    }else if (form.subj_ects.value == null ||
        form.subj_ects.value == undefined ||
        form.subj_ects.value.length == 0){
        alert("ECTS cannot be empty\n");
        return false;
    }else if (form.subj_semester.value == null ||
        form.subj_semester.value == undefined ||
        form.subj_semester.value.length == 0){
        alert("Semester cannot be empty\n");
        return false;
    }else {
        return true;
    }
}
</script>
```

Figura 57: Código JavaScript de comprobación de campos vacíos

4.4 Seguridad

Hoy en día es fundamental tener en cuenta el nivel de seguridad que una aplicación web requiere, ya que se gestionan muchos datos, al estar normalmente conectados a una BD. Aunque conseguir un proyecto con una alta protección y seguridad no era la meta principal, se ha tratado de cuidar un mínimo el desarrollo de este aspecto. Se ha tratado en dos sectores: el primero en la BD, y el segundo en la aplicación con la diferenciación entre administradores y usuarios.

- Seguridad de la BD: se ha creado un usuario con contraseña, que junto con el db_admin, serán los únicos que podrán tener acceso a la BD. Si se deseara se podrían cambiar las credenciales del usuario en el fichero de creación de la BD y del usuario. Este usuario también se usa para cualquier conexión a la BD en el programa de la aplicación web.
- Seguridad de la aplicación web: como se comentaba a lo largo de la memoria y en detalle en el punto 1 de esta sección de desarrollo, existen dos tipos de usuarios: usuarios estándar y usuarios administradores. Se han explicado también sus capacidades, y podríamos llegar a la conclusión de que los administradores se encargan de gestionar la herramienta. Es por esto por lo que es fundamental que solo ellos puedan realizar esta gestión, y que nadie que no sea administrador sea capaz de ello. Como se comentó en la subsección 1 (Roles de usuario), tenemos una aplicación común subdividida en administradores y usuarios estándar. Pero esto sirve de poco si ambas están abiertas en el mismo puerto de conexión a internet, ya que un usuario solamente tendría que conocer o descubrir el enlace de la página de administradores, y podría ejercer como uno de ellos. Para que funcione correctamente esta división, debería ejecutarse la parte de administradores en un puerto privado o interno, de forma que los usuarios, que utilizan la aplicación de usuarios (sobre un puerto público) no puedan acceder a la primera. Llevando esto a cabo solo podrían visualizar registros y nunca modificarlos o introducir nuevos.

No se ha utilizado un sistema de usuarios y contraseñas en la propia aplicación web debido a que se puede gestionar correctamente con la configuración de los puertos. Además, una buena versión con usuarios y contraseñas debería utilizar datos internos de la UAM y era algo complicado y que no estaba al alcance de este proyecto. Tampoco se han usado métodos preestablecidos como registrarse con Google ya que es una aplicación orientada a miembros de la EPS UAM, los cuales usarían en todo caso el correo de la universidad para identificarse en una aplicación de este estilo.

5 Integración, pruebas y resultados

Para conseguir una herramienta digna de cumplir los requisitos que se buscaban, era fundamental realizar multitud de pruebas, asegurando el buen funcionamiento de esta. En esta sección se mencionarán en primer lugar algunas pruebas que se han realizado durante el proceso de creación, y posteriormente se expondrá una breve entrevista con un usuario en local, en concreto un futuro estudiante de Erasmus.

5.1 Pruebas técnicas

Durante la realización y desarrollo del proyecto se han llevado a cabo múltiples pruebas. Cuando se añadía una nueva funcionalidad se probaba de manera que se pudieran ir tachando objetivos de la lista. Así, a lo largo del proyecto se ha podido ir mejorando la herramienta considerablemente con respecto a cuando se empezó. Las principales pruebas incluían el rellenado y envío de los formularios y la recepción de estos, la visualización correcta de los datos, la redirección correcta a otros ficheros y enlaces, así como el funcionamiento similar en cada ejecución en cuanto a tiempo.

También se simuló objetivamente el comportamiento de administradores y usuarios, suponiendo que los primeros podrían tener algún error en el rellenado de formularios. De esta manera se mejoró el control de estos errores, preparando a la herramienta para un comportamiento real. Simulando el comportamiento de un usuario se hizo más fácil el perfilado de la parte visual de la aplicación web, consiguiendo un aspecto mejor y más agradable y acogedor para la persona. También sirvió para mostrar los datos ordenados de una manera determinada y permitir al propio usuario la búsqueda y ordenación de registros.

Por último, una vez completado el desarrollo se procedió a realizar las pruebas finales, que involucraron la revisión de toda la herramienta, y la comprobación del correcto funcionamiento, incluso poniendo al límite la herramienta. Una de estas pruebas fue, por ejemplo, ejecutar rápido las acciones en la aplicación, comprobando que la herramienta funciona correctamente, aunque se someta a cierta presión. Otra de estas comprobaciones fue la ejecución e interacción con la aplicación por parte de un usuario, la cual se muestra en forma de entrevista en este próximo apartado.

5.2 Entrevista a futuro estudiante de Erasmus

En esta subsección se expondrá la breve entrevista realizada a un estudiante después de su interacción con la aplicación web como usuario estándar. Se llevó a cabo porque la opinión de un usuario, y más de un futuro estudiante de Erasmus, es una información útil para el desarrollador, ya que son los usuarios quienes mejor pueden decir lo que esperaban de la aplicación y qué tal han visto su diseño y funcionamiento. La entrevista con el estudiante, el cual ha preferido mantener como privada su identidad (solo cediendo como dato el detalle de que forma parte de la especialidad de la EPS de Sonido e Imagen), figura a continuación:

1. **¿Cómo has visto el diseño inicial de la aplicación? ¿Te parece cómodo el formato que se ha escogido de página inicial con todas las opciones en el menú desplegable?**

“El diseño es sencillo y es una aplicación que cualquier persona podría utilizar. Se ve claramente la utilidad para la que está destinada. El diseño es simple y podría mejorarse; tiene mucho hueco que puede hacer parecer un poco vacía la página.”

2. **¿Has entendido con facilidad el funcionamiento? ¿Crees que se podría hacer aún más sencillo?**

“Sí, lo he entendido correctamente. Es una aplicación que difícilmente puede hacerse más sencilla, ya que muestra directamente las opciones que te permite realizar.”

3. **Cuando procedes a visualizar los registros, ¿crees que se muestran de una forma correcta? ¿Cambiarías el formato de visualización de los datos?**

“Me ha gustado, porque se muestran de una manera limpia y ordenada, y además facilita el descargarlo en un formato específico, sin necesidad de estar viendo la página. Yo muchas veces prefiero descargarme un PDF, por ejemplo, en vez de estar consultando en internet.”

4. **Si has utilizado las opciones de ordenar y buscar, ¿te han parecido útiles? ¿Añadirías alguna más?**

“El buscador es bastante cómodo y puedes buscar por varios campos a la vez, lo cual es de gran ayuda, pero quizá estaría bien aplicar más filtros, como por ejemplo por departamentos o especialidad. La posibilidad de ordenar por columnas es muy útil también, pero podría ser más intuitivo que pulsando en una columna se ordena.”

5. **¿Has conseguido encontrar los datos que buscabas?**

“Sí, gracias a poder filtrar por campos en el buscador y ordenar por año, he podido encontrar prácticamente todas las asignaturas de mi itinerario (Imagen y Sonido) que necesitaba convalidar. Ha sido rápido y he podido ahorrar algo de tiempo.”

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Gracias a todas las tecnologías de desarrollo existentes, ha sido posible la creación de una herramienta para identificar convalidaciones de Erasmus, y así facilitar la tarea de búsqueda a los estudiantes. Algo que comenzó siendo una idea feliz debido a mi experiencia personal y la ardua tarea que me supuso el encontrar un destino para poder llevar a cabo mi Erasmus, acabó siendo un proyecto completo para cerrar una etapa y un reto personal para prepararme para lo que venga a continuación.

En este proyecto se ha logrado el objetivo de realizar una herramienta consistente en una aplicación web con diversas funcionalidades conectada a una BD, que pudiera separar los dos tipos de usuarios y que funcionase correctamente a nivel local, aunque se podría probar también remotamente. Todo ello ha sido posible gracias al seguimiento de varios pasos. En primer lugar, el estudio de las tecnologías a utilizar ha sido algo fundamental, requiriendo una gran investigación y aprendizaje previos al comienzo del diseño del proyecto. Una vez escogidas las tecnologías que se llevarían a cabo, se procedió al diseño de la BD, en conjunto con el de la aplicación web, para posteriormente comenzar a elaborar ambas secciones, así como la interconexión entre ellas. Se ha ido mejorando la herramienta desde una versión inicial, corrigiendo errores, mejorando la estructura de la BD y la parte de Frontend y Backend Web, leyendo e importando a la BD los ficheros Excel de la web de la EPS...; hasta finalmente conseguir obtener el proyecto compacto que se presenta en esta memoria.

Gracias al seguimiento de estos pasos se han logrado todos los objetivos que se buscaban tanto a nivel técnico como personal, y la experiencia ha sido en general satisfactoria a la par que costosa. Sin embargo, animo a todo el mundo que tenga una idea a creer en ella para algún día poder llegar a desarrollarla.

6.2 Trabajo futuro

Gracias a la opinión del estudiante entrevistado, la de mi tutor, y la del alumno y creador, se ha llegado a la conclusión de que habría varios aspectos posibles a mejorar en la herramienta:

- Probar la ejecución de la aplicación de forma remota, llevando a cabo los pasos requeridos.
- Exportar los datos a la nube, de forma que no se utilice apenas espacio en el servidor. No era algo necesario para esta herramienta al no manejarse una infinidad de datos, pero la nube es una tecnología en ascenso y cada día más popular.
- Diseñar de una manera más profesional la parte de Frontend de la aplicación web. Al ser una aplicación que se conecta a una BD, se ha dedicado más tiempo a perfeccionar esa parte que a la del aspecto visual, y aunque se ha tratado de cuidar el diseño, podría tener un aspecto más digno añadiendo algunas mejoras.
- Quizá se podría mejorar la usabilidad de las operaciones que involucren convalidaciones de varias asignaturas.
- Implementar un mayor nivel de seguridad para la herramienta.

Referencias

- [1] Universidad de Murcia “Capítulo 9. Sistemas de gestión de Bases de datos y SIG” https://www.um.es/geograf/sigmur/sigpdf/temario_9.pdf
- [2] MySQL Página oficial “MySQL Reference 5.0 Manual”
- [3] Software Testing Help “Top 10 free database software for Windows, Linux and Mac” <https://www.softwaretestinghelp.com/free-database-software/>
- [4] Edoc, Edgar “Historia de Oracle Corporation” <https://edoc.pub/historia-de-oracle-corporation-pdf-free.html>
- [5] Hostinger Tutoriales “MySQL” <https://www.hostinger.es/tutoriales/que-es-mysql/>
- [6] Tech Target “Microsoft SQL Server” <https://searchsqlserver.techtarget.com/definition/SQL-Server>
- [7] Database of Databases “Altibase” <https://dbdb.io/db/altibase>
- [8] Adelat “Lenguajes del lado servidor o cliente” (esquema) http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html
- [9] Sergio Luján Mora “Programación de aplicaciones web: historia, principios básicos y clientes web”, 2002, pp. 48-49.
- [10] Nicosiored “Tutorial ASP.Net en español – Introducción” <https://youtu.be/mYIJxJRafCo>
- [11] Guíadev “PHP vs JAVA” https://guiadev.com/php-vs-java/#Desarrollo_Backend
- [12] Juan Carlos García Candela “Introducción a JSP”, octubre 2004.
- [13] Oracle Sitio web oficial, Qusay H.Mahmoud “Web application development with JSP and XML” June 2001 <https://www.oracle.com/technical-resources/articles/javase/webappdev.html>
- [14] Píldoras informáticas “Curso Java Servlets I. Vídeo 246” https://www.youtube.com/watch?v=CZzyi_0ImwI
- [15] Blog Hackr “Python vs PHP” <https://hackr.io/blog/python-vs-php>
- [16] BitHeads Inc. “A Backend-as-a-service (BAAS) overview” <https://telusdigital-marketplace-production.s3.amazonaws.com/iot/user-content/product/818d-o.pdf>
- [17] J.D.Gauchat “El gran libro de HTML5, CSS3 y JavaScript”, 2012, pp. 31-45, 87, 101.
- [18] John Freddy Vega, Christian van Der Henst “Guía HTML5. El presente de la web. HTML5, CSS3 y JavaScript” (Traducido), Junio 2011
- [19] New Relic “10 amazing libraries and frameworks for your web-based JavaScript Applications” <https://blog.newrelic.com/engineering/best-javascript-libraries-frameworks/>
- [20] José Luis Comesaña “Manual básico de jQuery” https://www.sitiolibre.com/curso/pdf/Manual_jQuery.pdf
- [21] Sitio web oficial de jQuery-UI <https://jqueryui.com/>
- [22] Blog Hackr “Top JavaScript Libraries” <https://hackr.io/blog/top-javascript-libraries>
- [23] Píldoras informáticas “Curso JavaScript desde 0. Ajax I. Vídeo 68” <https://youtu.be/7w5PpdHKndc>
- [24] Víctor Robles Forcada “Programación web (Diseño de Servicios web)”, 2001
- [25] Juan Pavón “Acceso a bases de datos con JDBC”, 2013
- [26] Ing. Patricia Acosta “MS Excel 2007 con Programación de Macros en VBA”, mayo 2010, pp. 4-16.

- [27] Excel y Más “Enviar Registros desde Excel a SQLServer | VBA Excel 2013 #59”
<https://youtu.be/37BFzPDgxN4>
- [28] Excel Campus “How to use VBA Macros to Copy Data to another Workbook”
<https://www.excelcampus.com/vba/copy-paste-another-workbook/>
- [29] TheSpreadSheetGuru “Loop through all worksheets in all currently open workbooks”
<https://www.thespreadsheetguru.com/the-code-vault/2014/6/11/loop-through-all-worksheets-in-all-currently-open-workbooks>
- [30] Excel How “How to unmerge cells and fill down values in Excel”
<https://www.excelhow.net/how-to-unmerge-cells-and-fill-down-values-in-excel.html>
- [31] Sergio Luján Mora “Programación de servidores web con CGI, SSI e IDC”,
noviembre 2001

Glosario

BD	Base de datos
SQL	Structured Query Language
HTML5	HyperText Markup Language (version 5)
CSS3	Cascading Style Sheet
JS	JavaScript
AJAX	Asynchronous JavaScript and XML
JSON	JavaScript Object Notation
JDBC	Java DataBase Connectivity
VBA	Microsoft Visual Basic For Applications
EPS	Escuela Politécnica Superior
UAM	Universidad Autónoma de Madrid
API	Application Programming Interface

Anexos



A Manual de instalación

Existen ciertos prerequisites previos a la instalación de la herramienta:










- Instalar Tomcat como servidor de aplicaciones
- Instalar MySQL como gestor de base de datos

Para realizar una prueba completa de la herramienta son necesarios varios pasos:

- Descarga del fichero comprimido con los archivos necesarios. Este fichero está disponible si se solicita mediante un correo a miguel.vila@estudiante.uam.es
- Una vez se reciba el fichero, descomprimirlo. Dentro hay dos directorios:

 apache-tomcat-9.0.30
 BBDD

- Primero se debe abrir el directorio *BBDD*. Contiene lo siguiente:

 Erasmus	Carpeta de archivos	
 0_notas.txt	Archivo TXT	2 KB
 1_create_db_user.bat	Archivo por lotes ...	1 KB
 1_create_db_user.sql	Archivo SQL	1 KB
 2_create_tables.bat	Archivo por lotes ...	1 KB
 2_create_tables.sql	Archivo SQL	2 KB
 3_load_subjects_from_Excel.xlsm	Hoja de cálculo ha...	32 KB
 4_load_tables.bat	Archivo por lotes ...	1 KB
 4_load_tables.sql	Archivo SQL	5 KB

El primer directorio *Erasmus* incluye los ficheros Excel de la EPS que serán cargados a la Base de datos. Se pueden incluir otros ficheros Excel de convalidaciones que se deseen cargar a la BBDD, pero siempre respetando el formato de los ya existentes en la web de la EPS.

En el fichero *notas.txt* aparecen indicaciones y links a las referencias utilizadas para algunos posibles fallos en caso de que se complicase la instalación.















Cada fichero de extensión *sql* contiene las queries necesarias para crear la base de datos, y llenarla con registros. Los ficheros *.bat* son simplemente un script para la ejecución de esos ficheros. A continuación, comenzamos la ejecución:

- Abrimos un cmd en esa ubicación y ejecutamos *1_create_db_user.bat* para crear la base de datos y un usuario con sus credenciales.
- Ejecutamos *2_create_tables.bat* para crear las tablas que la base de datos contendrá.
- Abrimos el fichero Excel *3_load_subjects_from_Excel.xlsm*. Gracias a este fichero llevamos a cabo la lectura de los ficheros Excel del directorio *Erasmus* y el almacenamiento de los registros en la base de datos. Para llevar a cabo esto, es necesario:

- Tener habilitado el uso de macros en Excel (Botón de office > opciones de Excel > Centro de confianza > Configuración del centro de confianza > Habilitar todas las macros).
- También es necesario habilitar ActiveX y Microsoft VBA (alt+F11 > Herramientas > Referencias > Visual Basic for Applications y Microsoft ActiveX for Data Objects).

Una vez sigamos estos pasos, ya podremos utilizar las macros. Se puede ver su código mediante alt+F11, y mediante alt+F8 se ejecutan. Con la segunda opción se abrirá una ventana indicando la lista de macros. Escogeremos LoopThroughFiles y pulsaremos Ejecutar. Comenzará entonces la ejecución. Habrá que ir pulsando cualquier tecla para ir recorriendo fichero a fichero y hoja a hoja cada archivo de la carpeta *Erasmus*, hasta llegar al último. Los datos se irán almacenando en la base de datos. Se ha hecho poco a poco con el método de ir pulsando en vez de todo automáticamente, para poder ir comprobando que la inserción es correcta.

- El otro directorio *apache-tomcat-9.0.30* contiene el Tomcat sobre el cual se despliega la aplicación. En su interior podemos encontrar lo siguiente:

 bin	Carpeta de archivos	
 conf	Carpeta de archivos	
 lib	Carpeta de archivos	
 logs	Carpeta de archivos	
 temp	Carpeta de archivos	
 webapps	Carpeta de archivos	
 work	Carpeta de archivos	
 BUILDING.txt	Archivo TXT	20 KB
 CONTRIBUTING.md	Archivo MD	6 KB
 LICENSE	Archivo	57 KB
 NOTICE	Archivo	3 KB
 README.md	Archivo MD	4 KB
 RELEASE-NOTES	Archivo	7 KB
 RUNNING.txt	Archivo TXT	17 KB

Nos importan los directorios *bin* y *webapps*. En este último estará el fichero ejecutable de la aplicación web (*Erasmus.war*). El directorio *bin* será el lugar desde el que se realiza la ejecución.

Para ejecutar la aplicación web, lanzamos desde un cmd desde el directorio *bin* el comando `catalina.bat start`. Una vez hecho esto, ya estará activo el Tomcat, contenedor de Servlets y “servidor” de la aplicación web.

Para abrir la aplicación web simplemente tendremos que ir a un navegador, e insertar una de estas dos URLs:

- http://localhost:8180/Erasmus/UAM_Admin.html para ejecutar la aplicación de administradores.
- <http://localhost:8180/Erasmus/index.html> para ejecutar la aplicación de usuarios estándar.

B Ficheros de parámetros

Para cambiar la configuración por defecto de ciertos parámetros como puertos del tomcat o de la base de datos, así como los datos de usuario o contraseña, habría que actualizarlo en el fichero que corresponda:

- Puertos y parámetros del Tomcat: *server.xml*.
- Usuario y contraseña de la base de datos:
 - Carga inicial: *1_create_db_user.sql*.
 - Código fuente: *DbConnection.java*, *context.xml*.
 - Excel VBA: *run()*.